

# **PERANCANGAN SISTEM KEAMANAN MOTOR DENGAN MENGUNAKAN *STATE MACHINE***

**SKRIPSI**

**KEMINATAN TEKNIK KOMPUTER**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

ARYCCA SEPTIAN MULYANA

NIM:125150301111045



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018**

# **PENGESAHAN**

PERANCANGAN SISTEM KEAMANAN MOTOR DENGAN MENGGUNAKAN *STATE  
MACHINE*  
PENELITIAN

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Arycca Septian Mulyana  
NIM: 125150301111045

Skripsi ini telah diuji dan dinyatakan lulus pada  
18 Januari 2018  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Wijaya Kurniawan, S.T., M.T.  
NIP: 19820125 201504 1 002

Gembong Edhi Setyawan, S.T., M.T.  
NIK: 201208 761201 1 001

Mengetahui  
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D.  
NIP: 19710518 200312 1 001

## **PERNYATAAN ORISINALITAS**

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah penelitian ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah penelitian ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia penelitian ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 2018

Arycca Septian Mulyana

NIM:125150301111045

## KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah memberikan rahmat, taufik dan hidayah-Nya sehingga laporan penelitian dengan judul “Perancangan Sistem Keamanan Motor Dengan Menggunakan *State Machine*” ini dapat terselesaikan. Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan banyak pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Ibu **Srie Ramsiah, S.E.** selaku orang tua, yang selalu memberikan doa, motivasi, kasih sayang, dukungan moril dan materil sebagai penyemangat dalam menyelesaikan skripsi ini.
2. Ibu **Sri Indah Rosiani dan Sri Rusmiani** selaku bude dari penulis, yang selalu memberikan doa, motivasi, kasih sayang, dukungan moril dan materil sebagai penyemangat dalam menyelesaikan skripsi ini.
3. Bapak **Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D.** selaku Dekan Fakultas Ilmu Komputer.
4. Bapak **Heru Nurwarsito, Ir., M.Kom.** selaku Wakil Dekan I Bidang Akademik Fakultas Ilmu Komputer.
5. Bapak **Tri Astoto Kurniawan, S.T., M.T., Ph.D.** selaku Ketua Jurusan Teknik Informatika.
6. Bapak **Sabriansyah Rizkiqa Akbar, S.T., M.Eng.** selaku Ketua Program Studi Teknik Komputer yang sudah banyak membantu.
7. Bapak **Wijaya Kurniawan, S.T., M.T.** selaku dosen pembimbing I yang dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
8. Bapak **Gembong Edhi Setyawan, S.T., M.T.** selaku dosen pembimbing II yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
9. Segenap Bapak Ibu dosen dan staf serta karyawan Fakultas Ilmu Komputer Universitas Brawijaya atas segenap ilmu pengetahuan yang diberikan.
10. **M. Kholis Fikri, S.Kom., Ponco Wiguna, Embris Nuresalandis, S.Kom., Aras Nizamul Aryo Anwar, M. Wingga Woggiasworo, Fauzi Awal Ramadhan, Samkhya Aparigraha, Jodie Putra Kahir, Poby Zaarifwandono, S.Kom., Ardy Frayogi, S.Kom., Siti Rizky Amalia, S.K.G., Rahmat Yanuar Putra** terima kasih untuk dukungan dalam penyelesaian skripsi ini serta seluruh teman-teman sistem komputer angkatan 2012. **Ade Andriani Butar Butar, S.S.** yang sudah dengan sabar memberikan dukungan, semangat, dan doa selama pengerjaan skripsi ini dan pihak yang tidak bisa disebutkan satu per satu.
11. **Alm. Dio Falagki** selaku saudara serta sahabat dari penulis, yang selalu menemani dan membantu penulis dalam segala kondisi yang ada, serta selalu memberi dukungan dan doa selama pengerjaan skripsi ini.

12. **Dhipa Fathurahman., Thalia Hana S.M., Devana Mozaik, Mustika Tri Cahyani, S.I.Kom, Nursyabira, S.T., Kukuh Bhaskara, Siti Arfiah Meisari, S.Ked., Novanda Nuraniodi** untuk dukungan yang terus di berikan dalam penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 18 Januari 2018

Penulis

[aryccasm@gmail.com](mailto:aryccasm@gmail.com)

## ABSTRAK

Berdasarkan data yang didapatkan dari BPS (Badan Pusat Statistik) mengenai kasus pencurian motor di Indonesia menunjukkan peningkatan pada setiap tahunnya. Sehingga membuat masyarakat menjadi lebih khawatir dengan keamanan kendaraan roda duanya. Sudah banyak sistem keamanan kendaraan roda dua yang telah beredar di Indonesia, namun harga yang ditawarkan relatif terbilang mahal, sehingga membuat masyarakat enggan untuk menggunakannya. Untuk itu, penulis telah membuat perancangan sistem keamanan motor dengan menggunakan *State machine* yang memanfaatkan beberapa sensor dan teknologi yang sudah ada. Sensor yang digunakan merupakan sensor *fingerprint* dan sensor *GPS*, ditambah dengan beberapa teknologi berupa pengiriman data menggunakan *SMS-Gateway*. Sensor *fingerprint* digunakan untuk mendeteksi sidik jari dari pengguna kendaraan, *GPS* sebagai pembacaan lokasi dan pemberitahuan ketika kendaraan roda dua berpindah tempat saat diparkir, dan *SMS-Gateway* sebagai media pengiriman data lokasi dan pemberitahuan perpindahan kendaraan roda dua. Dalam pembuatan programnya akan digunakan *State machine* sebagai metodenya. Finite state machine memiliki 3 elemen dasar, yaitu *State*(keadaan), *Event*(kejadian) dan *Action*(aksi). Berdasarkan hasil yang telah didapatkan dari pengujian pada sistem, dapat disimpulkan dengan adanya sistem ini tingkat keamanan dari motor dapat meningkat, serta waktu yang didapatkan dalam pengiriman dan penerimaan data relatif efisien atau singkat dan penggunaan state machine pada sistem dapat berjalan dengan baik.

**Kata kunci** – Keamanan kendaraan roda dua, *GPS*, *Sensor fingerprint*, *SMS-Gateway*, *State machine*

## ABSTRACT

*Based on data obtained from BPS (Statistics Indonesia) regarding motorcycle theft cases in Indonesia shows an increase in each year. So that makes people become more worried about the safety of two wheeled vehicles. There are many security systems of two-wheeled vehicles that have been circulating in Indonesia, but the price offered relatively expensive, thus making people reluctant to use it. To that end, the authors have made the design of motorcycle security system by using a State machine that utilizes multiple sensors and existing technologies. The sensor used is a fingerprint sensor and GPS sensor, coupled with some technology in the form of data transmission using SMS-Gateway. The fingerprint sensor is used to detect fingerprints from vehicle users, GPS as location readings and notices when two-wheeled vehicles move while parked, and SMS-Gateway as a location data transmission medium and two-wheel drive notification. In making the program will be used State machine as the method. Finite state machine has three basic elements, namely State, Event and Action. Based on the results obtained from testing on the system, it can be concluded with the existence of this system the safety level of the motor can increase, and the time obtained in the delivery and reception of data is relatively efficient or brief and the use of state machine on the system can run well.*

**Keywords** - Security of two-wheeled vehicles, GPS, fingerprint Sensor, SMS-Gateway, State machine

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI .....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN .....	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah .....	2
1.3 Tujuan .....	3
1.4 Manfaat.....	3
1.5 Batasan masalah .....	4
1.6 Sistematika pembahasan .....	4
BAB 2 LANDASAN KEPUSTAKAAN .....	6
2.1 Tinjauan Pustaka .....	6
2.2 Dasar Teori .....	7
2.2.1 Definisi Keamanan.....	7
2.2.2 Definisi Short Message Service .....	7
2.2.3 Finite <i>State</i> Machine .....	8
2.2.4 Arduino Uno .....	10
2.2.5 Android <i>operation system</i> .....	10
2.2.6 Sensor Sidik Jari .....	11
2.2.7 Module SIM900A.....	11
2.2.8 Module <i>GPS</i> SKM53 .....	12
2.2.9 Module Relay 1 channel.....	13
2.2.10 LCD 16X4 .....	13



BAB 3 METODOLOGI .....	15
3.1 Metodologi Penelitian .....	15
3.1.1 Studi Literatur .....	16
3.1.2 Rekayasa Kebutuhan .....	16
3.1.3 Perancangan dan Implementasi Sistem.....	17
3.1.4 Pengujian dan Analisis.....	18
3.1.5 Kesimpulan .....	18
BAB 4 REKAYASA KEBUTUHAN .....	19
4.1 Rekayasa Kebutuhan Umum Sistem .....	19
4.1.1 Tujuan.....	19
4.1.2 Fungsi .....	19
4.1.3 Karakteristik Pengguna .....	19
4.1.4 Batasan Sistem .....	20
4.1.5 Asumsi dan Ketergantungan .....	20
4.2 Kebutuhan Antarmuka.....	20
4.2.1 Kebutuhan Antarmuka Pengguna .....	20
4.2.2 Antarmuka Perangkat Keras .....	20
4.2.3 Antarmuka Perangkat Lunak.....	21
4.3 Kebutuhan Fungsional .....	21
4.4 Kebutuhan Non-Fungsional .....	22
BAB 5 PERANCANGAN DAN IMPLEMENTASI .....	23
5.1 Gambaran Umum Sistem.....	23
5.2 Perancangan Sistem.....	24
5.2.1 Perancangan <i>State Machine</i> Pada Sistem Keamanan Motor .....	24
5.2.2 Perancangan <i>Frame</i> Sistem Keamanan Motor .....	25
5.2.3 Perancangan Aplikasi Menggunakan <i>ApplInverter</i> .....	29
5.3 Implementasi Sistem .....	31
5.3.1 Implementasi <i>State Machine</i> Pada Sistem Keamanan Motor ....	31
5.3.2 Implementasi <i>Frame</i> Sistem Keamanan Motor .....	35
5.3.3 Implementasi Aplikasi Menggunakan <i>ApplInverter</i> .....	42
BAB 6 PENGUJIAN DAN ANALISIS.....	47
6.1 Pengujian <i>State</i> Sistem Keamanan Motor .....	47

6.1.2 Pengujian <i>State</i> A : Mesin mati .....	47
6.1.3 Pengujian <i>State</i> B1 : Cek <i>GPS</i> .....	52
6.1.4 Pengujian <i>State</i> B2 : Cek <i>GPS</i> .....	53
6.1.5 Pengujian <i>State</i> C2 : Kirim Notif Lokasi .....	55
6.1.6 Pengujian <i>State</i> D : Cek Fingerprint .....	56
6.1.7 Pengujian <i>State</i> E : Mesin Nyala.....	58
6.1.8 Pengujian <i>State</i> B3 : Cek <i>GPS</i> .....	61
6.2 Analisis Pengujian .....	62
BAB 7 PENUTUP .....	64
7.1 Kesimpulan.....	64
7.2 Saran .....	64
DAFTAR PUSTAKA.....	65
LAMPIRAN A <i>Source Code</i> Program .....	66
A.1 Program Utama .....	66
A.2 Program Aktuator .....	68
A.3 Program <i>GPS</i> .....	68
A.4 Program <i>State</i> Machine.....	70
A.5 Program Sim900A .....	70
A.6 Program Fingerprint .....	71

## DAFTAR TABEL

Table 3.1 Kebutuhan perangkat keras .....	17
Table 5.1 Potongan Program <i>State Machine</i> .....	32
Table 6.1 Prosedur Pengujian Sistem Keamanan Motor .....	47
Table 6.2 Hasil Pengujian Aksi <i>Command</i> Memanggil Lokasi .....	48
Table 6.3 Hasil Pengujian Aksi time out 30 s .....	49
Table 6.4 Hasil Pengujian Pemindaian Sidik Jari .....	50
Table 6.5 Hasil Pengujian Aksi <i>Command</i> untuk Menyalakan Motor.....	51
Table 6.6 Hasil pengujian dengan time out 1s .....	52
Table 6.7 Hasil Pengujian Ketika Posisi Motor Berubah .....	54
Table 6.8 Hasi Pengujian Ketika Posisi Motor Tidak Berubah.....	54
Table 6.9 Hasil Pengujian aksi time out 30 s .....	56
Table 6.10 Hasil Pengujian Sidik Jari Pengguna Benar .....	57
Table 6.11 Hasil Pengujian Sidik Jari Pengguna Salah .....	57
Table 6.12 Hasil Pengujian <i>Command</i> Cek <i>GPS</i> .....	59
Table 6.13 Hasil Pengujian <i>Command</i> Mematikan Mesin .....	60
Table 6.14 Hasil Pengujian Cek <i>GPS</i> Dengan Time Out 1s .....	61

## DAFTAR GAMBAR

Gambar 2.1 Contoh Diagram Sederhana <i>State Machine</i> .....	8
Gambar 2.2 Arduino Uno .....	10
Gambar 2.3 Sensor Sidik Jari .....	11
Gambar 2.4 Module SIM900A.....	12
Gambar 2.5 Module <i>GPS</i> SKM53.....	12
Gambar 2.6 Module Relay 1 Channel .....	13
Gambar 2.7 LCD 16x4.....	14
Gambar 3.1 Metode Penelitian <i>Waterfall</i> .....	15
Gambar 3.2 Metode penelitian yang digunakan .....	16
Gambar 3.3 Diagram Blok Sistem .....	17
Gambar 5.1 Diagram Blok Sistem Keamanan Motor .....	23
Gambar 5.2 Perancangan Sistem .....	24
Gambar 5.3 Diagram Blok State Machine Pada Sistem Keamanan Motor .....	25
Gambar 5.4 Perancangan Pembacaan Module <i>GPS</i> SKM53.....	26
Gambar 5.5 Perancangan Module SIM900A.....	27
Gambar 5.6 Perancangan Sensor <i>Fingerprint</i> .....	28
Gambar 5.7 Perancangan Module Relay 1 Channel .....	29
Gambar 5.8 Tampilan Awal App Inverter.....	30
Gambar 5.9 <i>Flowchart</i> Perancangan Aplikasi Pada <i>Smartphone</i> .....	31
Gambar 5.10 Skema Pembuatan Perangkat Keras .....	36
Gambar 5.11 Bentuk Fisik Sistem Keamanan Motor .....	37
Gambar 5.12 Implementasi Sistem Keamanan Motor .....	38
Gambar 5.13 Implementasi <i>GPS</i> SKM53 .....	39
Gambar 5.14 Implementasi SIM900A .....	40
Gambar 5.15 Implementasi Sensor <i>fingerprint</i> .....	41
Gambar 5.16 Implementasi Relay 1 Channel.....	42
Gambar 5.17 Block Coding Speech Recognizer.....	42
Gambar 5.18 Block Coding Text and Sending Message Process.....	43
Gambar 5.19 Tampilan Aplikasi pada <i>Smartphone</i> .....	44
Gambar 5.20 Tampilan Aplikasi Ketika Menunggu Input Suara dari Pengguna ...	44

Gambar 5.21 Tampilan Aplikasi Ketika Pengguna Telah Meng-inputkan Suara...	45
Gambar 5.22 Tampilan Aplikasi Ketika Data Ditampilkan dan Dikirim.....	46
Gambar 6.1 Lokasi yang Benar.....	49
Gambar 6.2 Lokasi Pengujian ke – 4 .....	49
Gambar 6.3 Lokasi Pengujian ke – 7 .....	49
Gambar 6.4 Koordinat dari pengujian <i>State A</i> ke <i>State B2</i> .....	50
Gambar 6.5 Lokasi Hasil Pengujian <i>State B2</i> Aksi dua .....	55
Gambar 6.6 Lokasi Berdasarkan Data Dari Tabel 6.11.....	60

## DAFTAR LAMPIRAN

LAMPIRAN A <i>Source Code</i> Program .....	66
A.1 Program Utama .....	66
A.2 Program Aktuator .....	68
A.3 Program <i>GPS</i> .....	68
A.4 Program <i>State Machine</i> .....	70
A.5 Program Sim900A .....	70
A.6 Program Fingerprint .....	71

# **BAB 1 PENDAHULUAN**

## **1.1 Latar belakang**

Perkembangan teknologi pada era digital seperti sekarang ini sudah sangat banyak mengalami perubahan. Tidak hanya pada bidang teknologi itu sendiri, melainkan sudah mencakup bidang lainnya, salah satunya perkembangan pada bidang transportasi seperti kendaraan bermotor. Kendaraan bermotor sudah menjadi kebutuhan yang diperlukan masyarakat sehari-hari. Kendaraan bermotor dapat di bedakan berdasarkan kebutuhan yang digunakan oleh masyarakat, salah satunya adalah kendaraan bermotor roda dua. Kendaraan bermotor roda dua sudah menjadi sarana prasarana yang digunakan masyarakat untuk berpindah dari suatu tempat ke tempat lainnya, kendaraan bermotor roda dua dipilih dan banyak digunakan oleh masyarakat Indonesia karena harganya yang dapat dijangkau semua kalangan dan menjadi salah satu jenis transportasi yang dapat menghindari kemacetan-kemacetan yang sering terjadi di Indonesia.

Semakin banyaknya kendaraan yang digunakan oleh masyarakat maka sebanding dengan tingkat kejahatan pencurian kendaraan pribadi, terutama kendaraan roda dua atau motor. Berdasarkan data yang diperoleh dari BPS dan Biro Pengendalian Operasi tahun 2014 dapat dilihat bahwa kejahatan pencurian kendaraan bermotor tiap tahunnya selalu meningkat. Pada tahun 2011 telah terjadi 39.217 kasus pencurian kendaraan bermotor, pada tahun 2012 terjadi 41.816 kasus pencurian kendaraan bermotor dan pada tahun 2013 terjadi 42.508 kasus pencurian kendaraan bermotor (Statistik, 2014).

Melihat dari data tersebut, dapat disimpulkan bahwa kejahatan pencurian kendaraan bermotor ini harus diwaspadai, apalagi banyaknya kasus yang terjadi di kota – kota besar membuat resah masyarakat dan terlebih lagi buat para pelajar dan mahasiswa yang sedang menempuh pendidikannya.

Salah satu contoh kasus pencurian kendaraan roda dua dapat terjadi tidak hanya pada tempat umum melainkan juga dapat terjadi pada tempat-tempat yang memiliki pengamanan kendaraan bermotor terbilang aman seperti pada wilayah universitas. Pemilik kendaraan sepeda motor biasanya sudah menambahkan pengamanan tambahan seperti gembok, alarm, dan sebagainya, namun pencuri kendaraan roda dua masih dapat membawa kabur kendaraan roda dua walaupun sudah di lengkapi dengan pengamanan tambahan. Sebenarnya tindakan pertama dengan metode manual seperti menghubungi pihak berwajib untuk kasus kendaraan bermotor biasanya membutuhkan waktu proses penindakan yang cukup lama, padahal jika masyarakat dapat memanfaatkan teknologi yang telah berkembang dapat mempermudah dalam menangani kasus pencurian kendaraan roda dua. Salah satu contoh teknologi yang bisa digunakan untuk menangani kasus pencurian ini ialah penerapan sistem pelacakan dengan menggunakan *GPS*.

Berdasarkan dari permasalahan yang telah dijabarkan di atas, maka penulis ingin memanfaatkan teknologi *GPS* sebagai sistem utama pada keamanan

kendaraan roda dua yang berguna sebagai pemantau lokasi kendaraan roda dua dan memanfaatkan sensor pemindaian sidik jari untuk mengamankan kendaraan roda dua. Dengan menggunakan sensor sidik jari pada kendaraan dapat meningkatkan keamanan dari kendaraan tersebut. Namun jika dengan adanya keamanan pada kendaraan yang memanfaatkan sensor sidik jari masih dapat dicuri maka akan digunakan teknologi *GPS* untuk memantau lokasi motor yang telah dicuri, lokasi yang dilacak mencakup daerah atau wilayah yang luas. Sistem keamanan kendaraan roda dua ini akan dibuat lebih mudah untuk digunakan oleh pengguna ketika melacak lokasi dari kendaraan roda dua yang berupa koordinat longitude dan latitude yang dikirim dan diproses oleh mikrokontroler arduino. Penelitian terkait dengan bahasan penulis adalah “Rancang Bangun Sistem Keamanan Kendaraan Bermotor Berbasis *GPS*” (Baradja, 2016) dan “*GSM and GPS Based Vehicle Location and Tracking System*” (Kodavati, et al., 2012). Namun, dari dua penelitian yang menjadi tinjauan pustaka dari bahasan ini memiliki kelemahan berupa tidak adanya sistem pengamanan pertama pada kendaraan bermotor berupa sistem *authentication user* berupa pemindaian sidik jari dan tidak adanya pemberitahuan ketika kendaraan roda dua mengalami perubahan lokasi dari lokasi awal ke lokasi baru.

Pada penelitian yang dikerjakan oleh penulis menggunakan *state machine*, dikarenakan sistem keamanan motor yang penulis kerjakan memiliki banyak kondisi. Sehingga dengan penggunaannya *state machine* dapat memperingan komputasi yang ada, serta mampu memprediksi respon dari sistem berikutnya. Dan dengan adanya *state machine* ini mampu mengubah abstrak dari sistem menjadi kode program.

Oleh karena itu, dilihat dari kurang penelitian sebelumnya penulis ingin membuat sebuah sistem keamanan kendaraan roda dua dengan memanfaatkan *GPS* tidak hanya sebagai pemantau lokasi motor, namun dapat menjadi sistem pemberitahuan untuk pemilik kendaraan roda dua ketika kendaraan roda dua telah terparkir dan mengalami perpindahan lokasi. Selain penggunaan *GPS*, pada penelitian ini akan memanfaatkan teknologi *sms gateway*, pemindaian sidik jari dan aplikasi pada *smartphone* untuk memudahkan pengguna menggunakan sistem keamanan kendaraan roda dua ini dan menggunakan metode *state machine* dalam program sistemnya. Yang diharapkan dapat lebih membantu para pemilik kendaraan bermotor dalam menjaga keamanannya.

## 1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dibahas, dapat diambil beberapa rumusan masalah yang dapat dikaji sebagai berikut:

1. Bagaimana merancang dan mengimplementasikan sistem keamanan motor dengan menggunakan sistem keamanan motor berdasarkan sensor *fingerprint*, *GPS*, dan *SMS-Gateway*?
2. Berapa tingkat kecepatan waktu pengiriman dan menerima data lokasi pada sistem keamanan motor menggunakan *GPS* dan *SMS-Gateway*?



3. Bagaimana tingkat keberhasilan sistem dengan menerapkan *State Machine* dalam menjalankan sensor *fingerprint*, *GPS*, dan *SMS-Gateway* untuk sistem keamanan motor?

### 1.3 Tujuan

Tujuan yang diharapkan dari penelitian ini adalah:

1. Dapat merancang dan mengimplementasikan sistem keamanan pada kendaraan bermotor dengan menggunakan sistem keamanan motor berdasarkan sensor *fingerprint*, *GPS*, dan *SMS-Gateway*.
2. Dapat mengetahui tingkat kecepatan waktu pengiriman dan penerimaan data lokasi pada sistem keamanan motor menggunakan *GPS* dan *SMS-Gateway*.
3. Mengetahui tingkat keberhasilan sistem dengan menerapkan *State Machine* dalam menjalankan sensor *fingerprint*, *GPS*, dan *SMS-Gateway* untuk sistem keamanan motor.

### 1.4 Manfaat

Pada penelitian ini diterapkan sensor suara, *SMS Gateway*, dan juga *Smartphone* berbasis *android* untuk keamanan kendaraan sepeda motor, diharapkan dengan penerapan teknologi ini dapat diperoleh manfaat sebagai berikut:

#### a) Bagi penulis

1. Dengan penelitian ini, penulis dapat menerapkan ilmu yang telah dipelajari selama masa kuliah di Universitas Brawijaya.
2. Dapat memahami tentang penggunaan *State machine* serta dapat memahami dan mengetahui pengembangan perangkat tersebut pada bidang lainnya.
3. Dapat memahami tentang proses kerja pada arduino uno, penerapan modul *GPS*, modul *SIM900A*, dan sensor sidik jari, serta dapat memahami dan mengetahui pengembangan perangkat tersebut pada bidang lainnya.

#### b) Bagi pengguna

1. Meningkatkan keamanan pada sepeda motor, sehingga dapat mengurangi kejahatan pencurian sepeda motor
2. Memudahkan pengguna dalam memeriksa kendaraannya, karena telah diterapkannya *SMS Gateway* untuk *notifikasi* posisi sepeda motor
3. Dapat memanfaatkan *Smartphone* untuk menjaga keamanan sepeda motor.

Selain dari yang telah dijabarkan, penelitian ini dapat dijadikan rujukan – rujukan untuk penelitian sejenis, karena masih banyak bagian yang dapat dikembangkan dari penelitian ini.

### 1.5 Batasan masalah

Batasan – batasan masalah diperlukan agar dapat membantu penulis untuk memfokuskan penelitian ini supaya tidak menyimpang dari apa yang telah dituliskan pada rumusan masalah dan tujuan, maka penelitian ini memiliki batasan masalah sebagai berikut:

1. *Smartphone* yang digunakan berbasis *Android Lolipop 5.1.1*
2. Sistem mikrokontroller yang digunakan adalah *Arduino Uno*
3. Untuk pengiriman data melalui SMS Gateway menggunakan *SIM900A*.
4. Untuk pembacaan lokasi menggunakan modul *GPS SKM53*.
5. Untuk pembacaan atau pemindaian sidik jari menggunakan sensor *fingerprint*.
6. Menggunakan *State Machine*.

### 1.6 Sistematika pembahasan

Bagian ini berisi struktur penelitian ini mulai Bab Pendahuluan sampai Bab Penutup dan depenelitian singkat dari masing-masing bab. Diharapkan bagian ini dapat membantu pembaca dalam memahami sistematika pembahasan isi dalam penelitian ini.

**Bab I : Pendahuluan**

Mengurai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

**Bab II : Landasan Kepustakaan**

Menguraikan kajian pustaka dan dasar teori yang mendasari sistem keamanan motor menggunakan modul *GPS*, sidik jari dan sms gateway, serta *State machine*.

**Bab III : Metodologi**

Menguraikan tentang metode dan langkah kerja yang terdiri dari studi literatur, analisis kebutuhan simulasi, perancangan sistem, implementasi dan analisis serta pengambilan kesimpulan.

**Bab IV : Analisis dan rekayasa kebutuhan**

Menguraikan analisa dan rekayasa kebutuhan dari dasar teori yang telah dipelajari sesuai analisis.

**Bab V : Perancangan dan implementasi sistem**

Menguraikan proses perancangan dan implementasi dari dasar teori yang telah dipelajari sesuai analisis.

**Bab VI : Pengujian dan analisis**

Memuat hasil dan proses pengujian terhadap sistem keamanan motor menggunakan modul *GPS*, sidik jari dan sms gateway.

**Bab VII : Penutup**

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian program , serta saran-saran untuk mengembangkan lebih lanjut.

## BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepustakaan berisi uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan sistem keamanan motor menggunakan sidik jari dan sms gateway berbasis android menggunakan suara. Dalam landasan kepustakaan terdapat landasan teori dari berbagai sumber pustaka yang terkait dengan teori dan metode yang digunakan dalam penelitian. Jika dibutuhkan sesuai dengan karakteristik penelitiannya dan syarat kecukupan khusus keminatan tertentu, bisa juga terdapat kajian pustaka yang menjelaskan secara umum penelitian-penelitian terdahulu yang berhubungan dengan topik penelitian dan menunjukkan persamaan dan perbedaan penelitian tersebut terhadap penelitian terdahulu yang dituliskan.

### 2.1 Tinjauan Pustaka

Berikut ini beberapa penelitian terkait dengan sistem keamanan motor yang menggunakan beberapa sensor dan modul seperti *fingerprint*, *GPS*, dan *GSM* yang menjadi landasan dan referensi dari sistem keamanan motor yang akan dibuat.

Penelitian yang pertama, yaitu *An Intelligent Tracking System Based on GSM and GPS Using Smartphones*. Dengan menggunakan dua modul yaitu GSM dan GPS untuk membuat sebuah sistem keamanan motor. Dari kedua modul tersebut akan dibuat sistem *tracking* atau pelacakan pada kendaraan motor agar dapat meningkatkan keamanan motor tersebut. Namun, pada penelitian tersebut tidak terdapat sistem *notifikasi* atau pemberitahuan ketika motor telah berpindah tempat atau dicuri, serta tidak adanya metode yang digunakan pada sistem tersebut (Vigneshwaran, Sumithra, & Janani, 2015).

Penelitian kedua memaparkan tentang perancangan sebuah sistem *device switching* dengan menggunakan *GSM* dan *GPS*, serta ditambahkannya juga sebuah sensor *fingerprint*. Pada penelitian ini mengintegrasikan sensor *fingerprint* untuk meningkatkan keamanan pada sebuah *device* dan menggunakan sebuah LCD (*Liquid Crystal Display*) sebagai *interface* dari sistem. Namun pada penelitian ini hanya digunakan untuk melakukan *switching* pada *device* dan tidak diterapkannya sebagai sebuah sistem keamanan pada motor (Madhu, 2016).

Penelitian ketiga menjelaskan mengenai penerapan *state machine* pada implementasi *wireless sensor network* yang digunakan untuk mendukung pertanian modern. Pada penelitian ini lebih berfokus pada penerapan *WSN* untuk memonitoring tanaman sehingga dapat mempermudah petani dalam memantau kondisi tanaman karena data dikirim ke dalam *smartphone* dengan menggunakan jaringan *wireless* (Wibowo, 2018).

Berdasarkan penelitian yang telah dilakukan pada penelitian pertama dan kedua penulis mengambil tiga parameter, yaitu *GSM*, *GPS* dan sensor *fingerprint*, serta mengambil kelebihan-kekurangan penelitian satu dan dua

untuk membuat sistem baru yang lebih baik. Pada penelitian pertama, penulis mengembangkan dengan menambah sistem *notifikasi* atau pemberitahuan ketika motor telah berpindah tempat atau dicuri dan menambahkan juga sensor *fingerprint* seperti pada penelitian kedua. Pada penelitian kedua, penggunaan *GSM*, *GPS* dan sensor *fingerprint* akan diterapkan untuk sistem keamanan motor, serta menambahkan metode *State machine* sehingga dapat membuat sistem keamanan motor yang lebih baik. Serta berdasarkan penelitian ketiga penulis mengambil pemahaman mengenai cara membuat dan menggunakan *FSM (finite state machine)* pada sebuah sistem.

## **2.2 Dasar Teori**

Dasar teori berisi tentang teori-teori yang berhubungan dalam perancangan penelitian dengan topik rancang bangun pendeteksi kekeruhan air berbasis mikrokontroler pada penampungan air.

### **2.2.1 Definisi Keamanan**

Keamanan memiliki kata dasar pokok “aman” yang berarti bebas, terlindung dari bahaya, selamat, tidak membahayakan, yakin, dapat dipercaya, dapat diandalkan. Sedangkan “keamanan memiliki arti “suasana aman” ketenteraman, ketenangan (Salim & salim, 2002).

Dalam literatur kepolisian, pengertian keamanan secara umum adalah keadaan atau kondisi bebas dari gangguan fisik maupun Pshikis terlindunginya keselamatan jiwa dan terjaminnya harta benda dari segala macam ancaman gangguan dan bahaya.

Keamanan terklasifikasi menjadi 2 jenis keamanan, yaitu keamanan psikologis dan keamanan fisik. Keamanan psikologis dapat dikatakan sebagai keamanan yang cukup rumit untuk dipahami dan didapatkan oleh seseorang, dikarenakan untuk selamat dan aman secara psikologis, seseorang harus memahami apa yang diharapkan dari orang lain, termasuk anggota keluarga dan profesional pemberi perawatan kesehatan. Sedangkan, keamanan fisik melibatkan keadaan mengurangi atau mencegah ancaman pada tubuh dan kehidupan. Banyak ancaman yang dapat membahayakan seseorang dari bahaya yang sangat kecil hingga yang dapat menyebabkan kematian. Contohnya saja adalah penyakit, kecelakaan, bahaya pada lingkungan, termasuk pada kasus pencurian motor.

### **2.2.2 Definisi Short Message Service**

*Short Message Service (SMS)* merupakan layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel (nirkabel), memungkinkan dilakukannya pengiriman pesan dalam bentuk alphanumeric antar terminal pelanggan atau antar terminal pelanggan dengan sistem eksternal seperti *e-mail*, *paging*, *voice mail* dan lain-lain.

*Service SMS* membutuhkan sistem *SMS Center (SMSC)* yang menyimpan dan mem-forward text yang dikirimkan. Pada saat pesan SMS dikirim dari *handphone* pesan tersebut tidak langsung dikirimkan ke *handphone* tujuan, akan tetapi

dikirim terlebih dahulu ke *SMS Center* (SMSC), baru kemudian pesan tersebut diteruskan ke *handphone* tujuan.

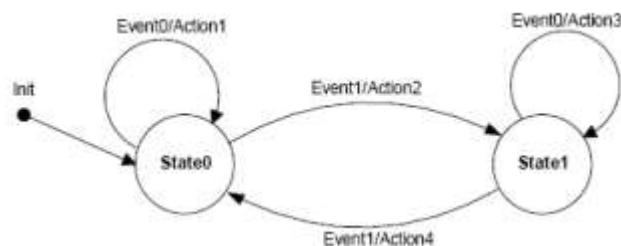
SMS Gateway adalah sebuah modul yang memungkinkan kita mengirim dan menerima pesan melalui SMS. Secara fungsional SMS gateway terpisah dari jaringan komunikasi bergerak, untuk itu SMS gateway dapat berinterkoneksi dengan semua platform teknologi informasi (IT). SMS gateway menyediakan interface-interface yang mudah digunakan (Easy to use) dan beberapa source platform yang sering digunakan seperti iSeries, Win32, Lotus Domino, Apache Web Server.

### 2.2.3 Finite State Machine

Finite State machine merupakan sebuah mesin yang dapat mengenali bahasa yang dipandang sebagai entitas-entitas abstrak, seperti sekumpulan string-string simbol alphabet tertentu.

#### 2.2.3.1 Defini Finite State Machine (FSM)

Secara umum, finite State machine adalah sebuah teknik atau metode yang digunakan untuk menggambarkan perilaku atau prinsip kerja pada sebuah sistem. Finite State machine memiliki 3 elemen dasar, yaitu *State*(keadaan), *Event*(kejadian) dan *Action*(aksi). Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu *State* yang aktif. Sistem dapat beralih atau bertransisi menuju *State* lain jika mendapatkan masukan atau event tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif kompleks (Setiawan, 2006).



**Gambar 2.1 Contoh Diagram Sederhana State Machine**

sumber: (Setiawan, 2006)

Pada gambar diagram di atas menunjukkan sebuah FSM(*Finite State Machine*) dengan dua buah *State*, satu buah input, empat buah *event/action* yang berbeda. Seperti yang terlihat pada gambar di atas, ketika sistem mulai dihidupkan, sistem akan bertransisi menuju *State0*, pada keadaan ini sistem akan menghasilkan *Action1* jika terjadi masukan *Event0*, sedangkan jika terjadi *Event1* maka *Action2* akan dieksekusi kemudian sistem selanjutnya bertransisi ke keadaan *State1* dan seterusnya.

*Event* merupakan sebuah kejadian yang dapat menyebabkan perpindahan dari satu *state* ke *state* yang lain ataupun melakukan pengulangan pada *state* yang sama. Terdapat dua jenis *event*, yaitu *event trasition* yang merupakan suatu kejadian yang dapat membuat perpindahan pada *state* dan *event iteration* yang merupakan suatu kejadian yang membuat *state* melakukan pengulangan. Sedangkan, *action* ditulis setelah nama *event* yang diawali dengan tanda garis miring (/). Karakteristik dari *action* harus bisa selesai sebelum *transition* mencapai *state* yang baru. Hal tersebut mengindikasikan bahwa *action* tidak dapat diinterupsi oleh *event* yang lain yang mungkin dideteksi oleh obyek.

FSM(*Finite State Machine*) didefinisikan sebagai pasangan 6 tupel  $F(K, V, S, Z, f, g)$  dimana:

- K : himpunan hingga stata
- $V_T$  : himpunan hingga simbol input (alphabet)
- $S \in K$  : stata awal
- Z : himpunan hingga simbol output
- $f : K \times V_T \rightarrow K$  disebut fungsi next *State*
- $g : K \times V_T \rightarrow Z$  disebut fungsi output

#### 2.2.3.2 Kelebihan dan Kekurangan FSM(*Finite State Machine*)

- Kelebihan *State machine*  
FSM memiliki beberapa kelebihan (Brownlee, 2010), diantaranya :
  1. Sederhana, sehingga mudah diimplementasikan
  2. Bisa diprediksi responnya
  3. Komputasi ringan
  4. Relatif fleksibel
  5. Merupakan metode AI lama yang bisa digunakan pada berbagai sistem
  6. Mudah ditransfer dari abstrak menjadi kode program
- Kekurangan *State machine*  
Selain memiliki banyak kelebihan, FSM juga mempunyai beberapa kelemahan (Brownlee, 2010),diantaranya :
  1. Karena sifatnya bisa diprediksi, maka implementasi pada game kurang disukai
  2. Implementasi pada sistem yang lebih besar lebih sulit karena pengaturan dan pemeliharaannya jadi kompleks

3. Sebaiknya hanya digunakan pada sistem dimana sifat sistem bisa didekomposisi menjadi *State*.
4. Kondisi untuk transisi *State* adalah tetap.

#### 2.2.4 Arduino Uno

Arduino Uno merupakan papan mikrokontroller berbasis Atmega32b(datasheet). Arduino uno termasuk dalam keluarga besar dari arduino yang merupakan sebuah *platform hardware* yang bersifat *open-source*. Bahasa pemrograman yang digunakan pada arduino memiliki kemiripan *syntax* dengan bahasa pemrograman C.

Arduino memiliki pin input sebanyak 14 pin, dimana dari 14 pin tersebut terbagi menjadi 6 pin input digunakan sebagai output PWM dan 6 pin input analog. Arduino uno juga memiliki 16MHz osilator kristal, koneksi USB(unit serial bus), jack power, ICSP header dan tombol reset.

Nama “Uno” sendiri berasal dari bahasa italia yang berarti satu, yang menandakan perilisan pertama Arduino software (IDE) 1.0. Arduino uno dan versi 1.0 dari arduino software (IDE) ini lah menjadi referensi dari versi – versi arduino berikutnya.



**Gambar 2.2 Arduino Uno**

sumber : (Arduino, 2017)

#### 2.2.5 Android operation system

Android adalah software untuk perangkat mobile yang mencakup sistem operasi, middleware dan aplikasi kunci. Pengembangan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Serangkaian aplikasi inti Android antara lain klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain.

Dengan menyediakan sebuah platform pengembangan yang terbuka, pengembang Android menawarkan kemampuan untuk membangun aplikasi yang sangat kaya dan inovatif. Pengembang bebas untuk mengambil keuntungan dari



perangkat keras, akses informasi lokasi, menjalankan background services, mengatur alarm, tambahkan pemberitahuan ke status bar, dan banyak lagi.

Android bergantung pada versi Linux 2.6 untuk layanan sistem inti seperti keamanan, manajemen memori, manajemen proses, network stack, dan model driver. Kernel juga bertindak sebagai lapisan abstraksi antara hardware dan seluruh software stack.

#### **2.2.6 Sensor Sidik Jari**

Sensor sidik jari merupakan sebuah alat yang dapat membaca dan menyimpan data berupa pola sidik jari manusia. Sebuah *fingerprint scanner* memiliki dua cara bekerja, yaitu mengambil gambar sidik jari pengguna dan memutuskan apakah pola alur sidik jari dari gambar yang diambil sama dengan pola alur sidik jari yang ada di database.

Menurut Henry Classification System, pada umumnya sidik jari dapat dibagi menjadi 3 jenis berdasarkan pola sidik jarinya, yaitu *loop pattern*, *whorl pattern*, dan *arch pattern*. Dengan menggunakan pola – pola tersebut sensor *fingerprint* dapat membaca dan memproses sidik jari manusia, serta membedakan sidik jari manusia. Dapat dilihat pada gambar di bawah ini bentuk fisik dari sensor sidik jari.



**Gambar 2.3 Sensor Sidik Jari**

sumber : (duwiarsana.com)

#### **2.2.7 Module SIM900A**

Module SIM900a merupakan sebuah modul yang dapat digunakan sebagai pengirim dan penerima pesan sms, serta dapat melakukan panggilan telpon. Module SIM900a juga merupakan salah satu produk berbasis GSM/GPRS dari SIMCOM yang dapat digunakan dengan sebuah mikrokontroler seperti arduino atau sebuah mikroprosesor seperti raspberry pi.

Module SIM900a menggunakan sebuah perintah pada penerapannya yaitu *AT+command*. Dengan menggunakan *AT+command*, pengguna dapat memberikan perintah, seperti mengirim dan menerima pesan. Dalam penerapan

AT+*command*, biasanya digunakan pada komunikasi serial, sehingga SIM900a merupakan module dengan komunikasi serial.



**Gambar 2.4 Module SIM900A**

sumber : (bukalapak.com)

#### **2.2.8 Module GPS SKM53**

*GPS (Global Positioning System)* merupakan sebuah sistem dengan memanfaatkan satelit sebagai navigasi, penentu lokasi dan keberadaan secara kontinyu. Pada umumnya *GPS* bekerja dengan cara memancarkan sinyal kemudian diterima oleh satelit di luar angkasa yang berikutnya akan ditransmisikan sinyal dari satelit ke perangkat *GPS*, seperti contohnya *smartphone* yang dilengkapi teknologi *GPS*.

Salah satu jenis modul *GPS* yang dapat digunakan yaitu modul *GPS SKM53*. *GPS SKM53* merupakan module embedded yang memiliki antenna *GPS* di dalamnya untuk memungkinkannya membaca lokasi. Module SKM53 ini dapat bekerja dengan baik jika langsung menghadap ke langit, sehingga module SKM53 ini tidak dapat digunakan dengan baik di dalam ruangan.



**Gambar 2.5 Module GPS SKM53**

sumber : (skylab.com.cn)

Module SKM53 memiliki fitur 3327 *single-chip* arsitektur dari MediaTek, dengan sensitivitas -165dBm sehingga cakupan areanya dapat meluas seperti daerah perkotaan dan lingkungan dengan perpohonan.

#### 2.2.9 Module Relay 1 channel

Modul relay merupakan perangkat elektronik yang memiliki sistem kontrol atau *switch* yang dapat digunakan untuk mengendalikan rangkaian arus tinggi dengan sinyal arus rendah. Sangat baik untuk melakukan *switch* pada perangkat AC maupun DC yang membutuhkan arus dan tegangan yang besar.



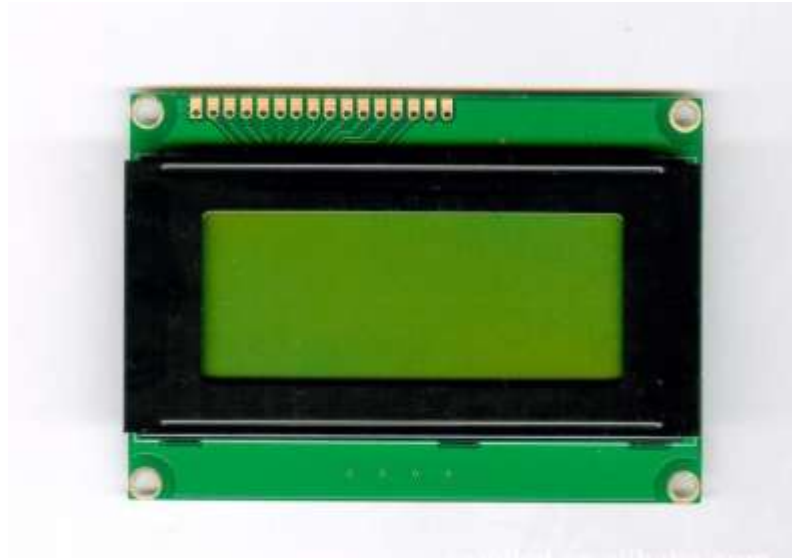
**Gambar 2.6 Module Relay 1 Channel**

sumber : (bukalapak.com)

#### 2.2.10 LCD 16X4

*Liquid Cristal Display* atau yang biasanya disingkat menjadi LCD merupakan salah satu jenis dari sebuah display elektronik yang memiliki fungsi sebagai tampilan dari suatu data, baik karakter, huruf, angka ataupun grafik. LCD merupakan lapisan yang terdiri dari lapisan campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan *seven-segment* dan lapisan elektroda pada kaca belakang.

Di dalam sebuah modul LCD (*Liquid Cristal Display*) terdapat mikrokontroler yang berfungsi sebagai pengendali tampilan karakter LCD (*Liquid Cristal Display*). Mikrokontroler pada sebuah LCD dilengkapi dengan memori dan register.



**Gambar 2.7 LCD 16x4**

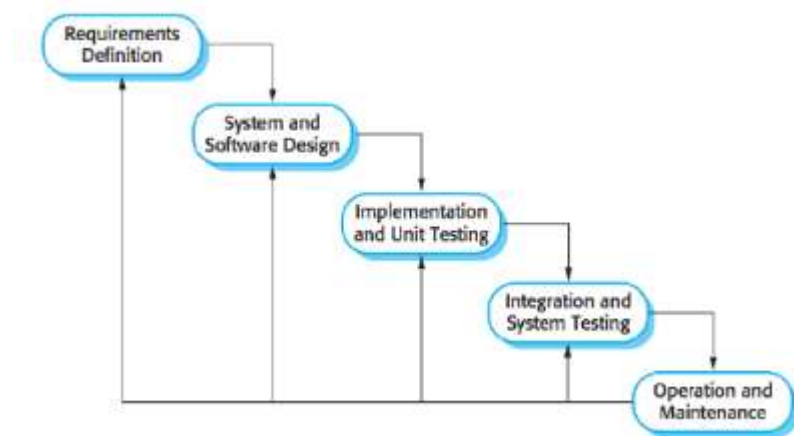
sumber: (AliExpress.com)

## BAB 3 METODOLOGI

Pada bab ini akan dijabarkan langkah – langkah yang akan dilakukan untuk menyelesaikan masalah penelitian ini. Langkah-langkah tersebut meliputi studi literatur, analisis kebutuhan, perancangan sistem, implementasi sistem, analisis hasil pengujian dan pengambilan kesimpulan.

### 3.1 Metodologi Penelitian

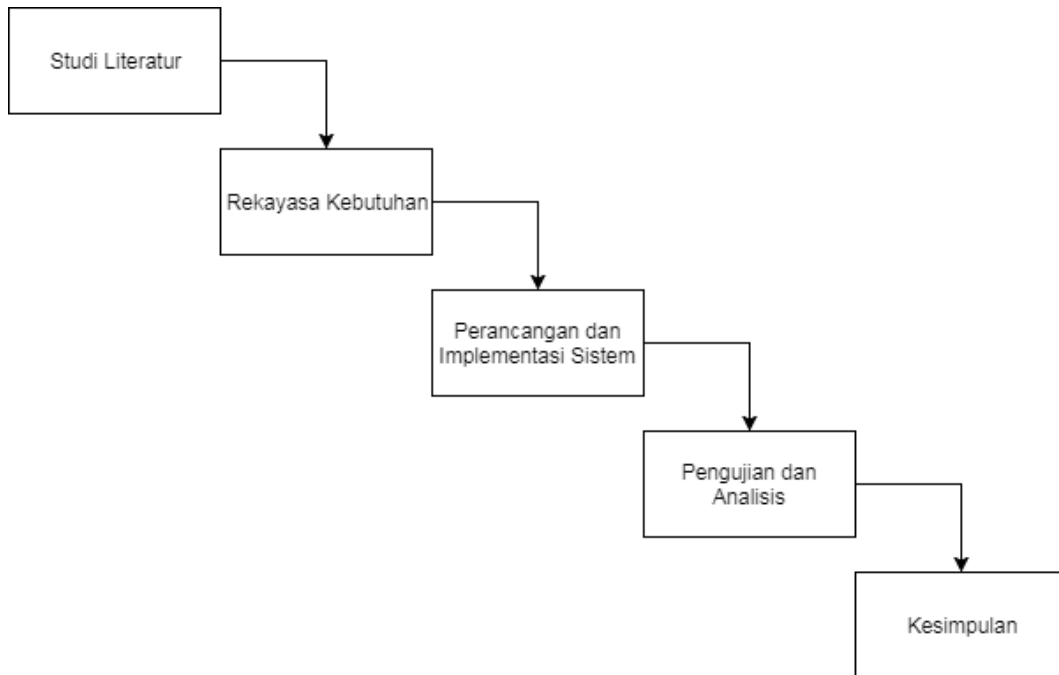
Metodologi penelitian yang akan digunakan pada penelitian ini yaitu metode penelitian *waterfall*. Metode ini merupakan metode pengembangan perangkat lunak terstruktur yang paling dikenal dan banyak digunakan secara luas, tidak hanya di lingkup akademisi tetapi juga di industri (Binanto, 2015).



**Gambar 3.1 Metode Penelitian *Waterfall***

sumber : (Sommerville, 2011)

Pada Gambar 3.1 merupakan metode penelitian *waterfall* versi Sommerville(Sommerville, 2011). Mengambil dasar dari metode penelitian tersebut penulis mengadaptasikannya pada metode penelitian penulis dengan menambahkan dan merubah beberapa bagian sehingga dapat disesuaikan dengan alir penelitian penulis yang ditunjukkan pada Gambar 3.2.



**Gambar 3.2 Metode penelitian yang digunakan**

### **3.1.1 Studi Literatur**

Studi literatur merupakan kegiatan yang dilakukan untuk mempelajari serta memahami penjelasan dasar teori yang digunakan untuk menunjang perancangan agar tidak mengalami kendala. Pada tahap studi literatur ini, mempelajari teori-teori yang digunakan dalam pengerjaan penelitian. Teori-teori pendukung tersebut diperoleh dari buku, jurnal, e-book, dokumentasi, dan penelitian sebelumnya yang berkaitan dengan penelitian. Referensi utama yang diperoleh dalam penulisan ini adalah forum-forum terkait dengan komponen dan metode *state machine* yang diterapkan dalam penelitian ini, seperti forum diskusi pengguna *arduino uno* dan jurnal.

### **3.1.2 Rekayasa Kebutuhan**

Rekayasa kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan dalam merancang sistem yang akan dibuat dan diuji. Analisa kebutuhan dilakukan dengan cara mengidentifikasi kebutuhan dari sistem dan peralatan yang terlibat didalamnya. Dalam kebutuhan sistem akan terjadi proses mengidentifikasi beberapa perangkat yang digunakan seperti perangkat keras dan perangkat lunak. Dengan adanya pengidentifikasi, maka dapat mempermudah dalam perancangan dan implementasi pada saat pembuatan sistem. Berikut adalah kebutuhan sistem pada perangkat keras dan perangkat lunak:

#### **1 Kebutuhan perangkat keras**

Kebutuhan perangkat keras yang digunakan pada penelitian ini akan dijelaskan pada tabel 3.1

**Table 3.1 Kebutuhan perangkat keras**

Nama perangkat	Kegunaan
Sensor Sidik Jari	Sebagai pemindai sidik jari pengguna
Arduino Uno	Sebagai Mikrokontroler
GPS53	Sebagai penunjuk lokasi
SIM900A	Sebagai pengirim data dari mikrokontroler
Relay 1 Channel	Sebagai saklar untuk menghidupkan dan mematikan motor
Smartphone Android	Sebagai pemberi perintah terhadap mikrokontroler

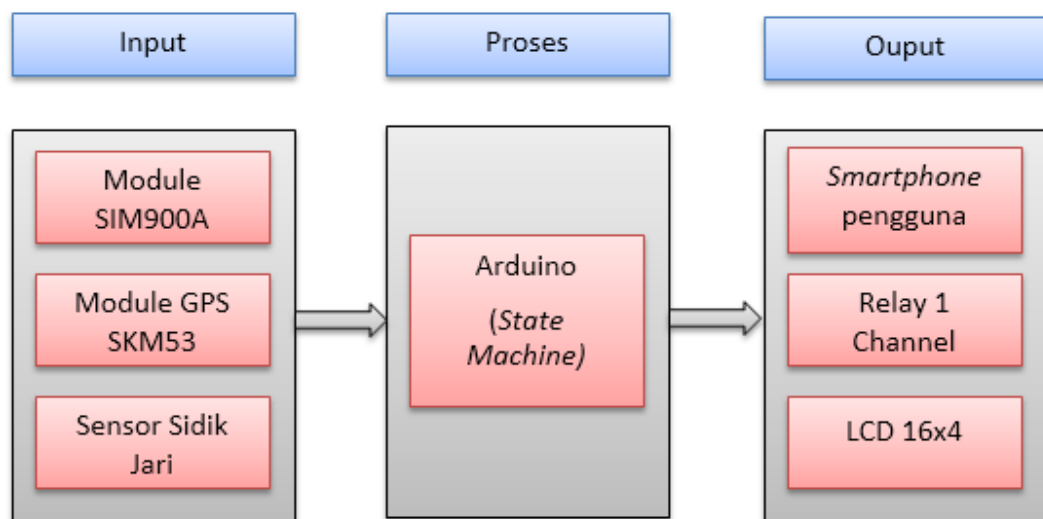
## 2 Kebutuhan perangkat lunak

Kebutuhan perangkat lunak yang digunakan pada penelitian ini akan dijabarkan sebagai berikut :

- a) Arduino Softward IDE
- b) *App inverter*

### 3.1.3 Perancangan dan Implementasi Sistem

Setelah melakukan rekayasa kebutuhan untuk sistem maka selanjutnya akan dilakukan perancangan serta implementasi pada sistem. Untuk mempermudah dalam memahami perancangan sistem akan digambarkan dalam bentuk diagram blok pada Gambar 3.3 yang akan menjelaskan desain sistem secara keseluruhan.



**Gambar 3.3 Diagram Blok Sistem**

Dari diagram blok pada Gambar 3.3, merupakan sebuah alur perangkat keras yang akan digunakan dalam penelitian berupa modul SIM900A, modul *GPS*, dan sensor sidik jari yang digunakan untuk membaca pesan perintah dari pengguna, membaca lokasi, dan membaca sidik jari dari pengguna yang kemudian akan diproses oleh arduino uno. Modul dan sensor yang digunakan pada sistem ini diletakkan di dalam kotak yang terbuat dari bahan akrilik, kemudian jika terdapat perintah yang masuk melalui pesan, perubahan lokasi pada alat dan adanya pemindaian sidik jari maka data akan diolah di dalam mikrokontroler arduino uno. Setelah diolah oleh arduino maka pesan perintah yang telah dikirimkan oleh pengguna akan dijalankan sesuai dengan keinginan pengguna, kemudian data lokasi yang berubah akan dikirimkan ke *smartphone* pengguna, serta data dari sensor sidik jari yang telah diolah akan membuat relay berfungsi dan relay akan menyalakan motor.

Implementasi sistem dilakukan sesuai dengan perancangan sistem yang telah dibuat sebelumnya. Pada bagian ini terdapat bermacam – macam proses implementasi, antara lain yaitu implementasi komunikasi menggunakan *sms-gateway*, implementasi pemindaian atau pembacaan sidik jari, implementasi pembacaan lokasi, serta implementasi untuk mematikan dan menyalakan motor.

#### **3.1.4 Pengujian dan Analisis**

Pengujian sistem pada penelitian ini dilakukan agar dapat menunjukan bahwa pengembangan dari sistem ini telah mampu bekerja sesuai dengan spesifikasi dari keutuhan yang melandasinya. Pengujian yang dilakukan meliputi:

1. Pengujian pada sensor sidik jari yang bertujuan untuk mendeteksi atau memindai sidik jari dari pengguna.
2. Pengujian pengiriman data dari pengguna ke arduino uno atau sistem dengan menggunakan komunikasi *sms-gateway*.
3. Pengujian pada modul *GPS* untuk menentukan data lokasi.
4. Pengujian pada modul *GPS* untuk membandingkan lokasi sekarang dengan lokasi sebelumnya.
5. Pengujian pada modul relay untuk menyalakan dan mematikan motor.

Setelah dilakukannya pengujian, maka akan dilakukan analisis terhadap pengujian tersebut dengan berdasarkan data yang telah didapat dari pengujian. Dengan begitu akan dapat diketahui mengenai kinerja dari sistem.

#### **3.1.5 Kesimpulan**

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan implementasi dan pengujian telah selesai. Kesimpulan diambil untuk menjawab rumusan masalah yang sudah dijabarkan sebelumnya. Tahapan terakhir dari penulis adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta memberikan pertimbangan yang terjadi atas hasil yang telah dilakukan.



## BAB 4 REKAYASA KEBUTUHAN

### 4.1 Rekayasa Kebutuhan Umum Sistem

Secara rekayasa kebutuhan umum sistem dibutuhkan beberapa hal yang diperlukan untuk melakukan perancangan sistem ini yaitu tujuan, fungsi, dan karakteristik pengguna.

#### 4.1.1 Tujuan

Tujuan dari perancangan sistem ini adalah untuk menguraikan dan menjelaskan bagaimana penelitian membangun sistem keamanan pada motor agar dapat mengurangi tingkat kejahatan terhadap pencurian kendaraan motor yang sering terjadi di kota-kota besar Indonesia.

Adapun penguraian yang lebih rinci pada penelitian ini adalah adalah fitur dari sistem yang dikembangkan, kegunaan, lingkungan pengguna, batasan perencanaan dan implementasi dan asumsi ketergantungan, serta penjelasan terkait kebutuhan fungsional dan non fungsional. Pada bab ini dibuat agar dapat digunakan untuk dokumentasi para pengembang berikutnya supaya selanjutnya penelitian ini dapat dikembangkan dan diusulkan sebagai laporan penelitian.

#### 4.1.2 Fungsi

Dalam perancangan sistem keamanan motor ini, akan dilengkapi dengan mikrokontroler, modul *GPS*, *SIM900*, dan sensor *fingerprint* pada kendaraan motornya. Kegunaan dari *SIM900* adalah untuk menerima dan mengirimkan data dari dan untuk pengguna atau *smartphone* yang telah terinstall aplikasi sebelumnya digunakan oleh pengguna ke mikrokontroler yang berada di motor. Kemudian, mikrokontroler yang berada pada motor berguna untuk menjalankan dan memproses modul serta sensor yang ada. Kegunaan pada modul *GPS* sebagai penunjuk lokasi motor kepada pengguna ketika pengguna ingin memastikan lokasi dari motor pengguna. Sedangkan sensor *fingerprint* berguna sebagai *authentication user*, apabila ketika yang menggunakan bukan pengguna dari motor tersebut, maka motor tidak dapat menyala walaupun telah di-*starter*, dan saat pengguna yang menggunakannya, maka dapat menyala dengan bantuan relay. Selain menggunakan sensor *fingerprint*, pengguna juga dapat melakukan perintah nyalakan dan matikan motor dengan bantuan suara pada *smartphone* yang digunakan. Suara akan diterima dan diproses pada *smartphone* kemudian akan dikirim ke mikrokontroler melalui komunikasi dengan *SIM900*.

#### 4.1.3 Karakteristik Pengguna

Karakteristik pengguna pada sistem ini bersifat aktif, karena pengguna dapat melakukan fungsi pada sistem dengan menggunakan aplikasi yang telah dibuat pada *smartphone*. Pengguna bersifat pasif ketika pengguna tidak menggunakan aplikasinya, namun tetap mendapatkan informasi lokasi dari sistem ketika lokasi pada sistem berubah.

#### 4.1.4 Batasan Sistem

Beberapa batasan yang ada pada sistem ini antara lain :

1. Alat tidak dapat digunakan ketika kondisi cuaca dalam keadaan hujan.
2. Sidik jari pengguna ketika akan menggunakan sensor sidik jari harus dalam keadaan bersih, tidak kotor dan basah.
3. Komunikasi *sms-gateway* memanfaatkan sinyal dari *provider*.
4. Modul *GPS* SKM53 tidak dapat memberikan informasi lokasi secara akurat jika berada di dalam ruangan.
5. Sistem ini menggunakan sumber daya luar dari aki atau power supply 12v.

#### 4.1.5 Asumsi dan Ketergantungan

Beberapa asumsi dan ketergantungan yang ada pada sistem ini antara lain :

1. Modul *GPS* akan membaca lokasi secara kontinyu.
2. Ketika sistem melakukan perbandingan antara lokasi 1 dengan lokasi 2 tidak ada perbedaan, sistem tidak akan mengirim pemberitahuan.
3. Sistem ini membutuhkan minimal tegangan 12v untuk menjalankan sistem.

### 4.2 Kebutuhan Antarmuka

Kebutuhan antarmuka merupakan kebutuhan untuk menghubungkan pengguna sistem dan sistem itu sendiri. Pada kebutuhan antarmuka ini terbagi menjadi 3, yaitu kebutuhan antarmuka pengguna, kebutuhan antarmuka perangkat keras dan kebutuhan antarmuka perangkat lunak.

#### 4.2.1 Kebutuhan Antarmuka Pengguna

Kebutuhan yang diperlukan untuk antarmuka pengguna dalam sistem ini adalah sebagai berikut:

1. Sistem dapat melakukan nyala dan mati pada kunci ganda, serta melakukan proses *tracking* terhadap lokasi motor dengan baik.

Sistem dapat terhubung dengan baik terhadap *smartphone* untuk melakukan proses nyala dan mati motor, serta melakukan proses *tracking*.

#### 4.2.2 Antarmuka Perangkat Keras

Kebutuhan perangkat keras yang dibutuhkan untuk antarmuka pada sistem ini yaitu:

1. Modul SIM900
  - Membutuhkan daya sebesar 5V.
  - Sebagai komunikasi pada sistem, yaitu penerima dan pengirim data melalui via sms.

## 2. Modul *GPS* SKM53

- Membutuhkan daya sebesar 5V.
- Sebagai penentu data lokasi pada motor.

## 3. Modul relay 1 channel

- Membutuhkan daya sebesar 5V.
- Sebagai pengaktif fungsi nyala dan mati pada motor.

## 4. Modul *fingerprint*

- Membutuhkan daya sebesar 5V.
- Sebagai pengenalan terhadap pemilik motor

### 4.2.3 Antarmuka Perangkat Lunak

Kebutuhan perangkat lunak yang dibutuhkan untuk antarmuka pada sistem ini yaitu:

#### 1. Modul SIM900

Untuk menghubungkan dan mengirimkan data dari dan untuk pengguna kepada mikrokontroler pada motor, dengan memanfaatkan aplikasi pada *smartphone* nantinya. Aplikasi tersebut yang akan digunakan untuk melakukan proses perintah nyala dan mati pada kunci ganda, serta *tracking* lokasi motor.

#### 2. *App inverter*

Untuk membuat aplikasi yang akan digunakan pada sistem ini. Aplikasi yang dibuat dapat digunakan pada *smartphone* dengan o.s. android.

### 4.3 Kebutuhan Fungsional

Dalam perancangan dan implementasi sebuah sistem dibutuhkan sebuah kebutuhan fungsional untuk menunjang kinerja dari sistem agar dapat bekerja dengan baik seperti yang diharapkan.

#### 1. Sistem dapat melakukan proses nyala dan mati kunci ganda motor dengan baik

##### a. Penjelasan

Fitur ini berguna untuk menjelaskan bagaimana sistem nyala dan mati pada motor berjalan. Agar sistem dapat melakukan proses tersebut, cara pertama pengguna hanya perlu menggunakan aplikasi yang telah terinstall di *smartphone* yang sebelumnya telah terhubung dengan modul SIM900 pada mikrokontroler. Pada aplikasi, pengguna hanya perlu memberi perintah dengan suara untuk menyalakan dan mematikan pada motor. Kemudian untuk cara kedua, pengguna

dapat menggunakan sidik jari pada modul *fingerprint* yang telah terpasang pada motor untuk menyalakan motor.

b. Stimulus dan respon sistem

Sistem akan menerima dan mengirim data yang ada dari mikrokontroler arduino kepada aplikasi yang sudah terinstall pada *smartphone* dengan menggunakan modul SIM900 sebagai jalur komunikasi.

c. Kebutuhan fungsional

Kebutuhan fungsional ini digunakan sebagai cara agar pengguna dapat melakukan proses hidup dan mati pada motor, supaya pengguna dapat merasa lebih aman ketika akan meninggalkan motornya.

2. Sistem dapat memberikan informasi lokasi motor dengan baik.

a. Penjelasan

Fitur ini berguna untuk menjelaskan modul *GPS SKM53* pada mikrokontroler dapat mengirimkan data lokasi motor yang akurat ketika pengguna ingin mengetahui lokasi dari kendaraan motor yang pengguna miliki.

b. Stimulus atau respon sistem

Sistem akan menerima data berupa lokasi dari mikrokontroler arduino ke aplikasi yang sudah terinstall pada *smartphone* dengan data dari modul SKM53 sebagai acuan data yang akurat.

c. Kebutuhan fungsional

Kebutuhan fungsional ini digunakan sebagai cara agar pengguna dapat mengetahui lokasi dari motornya pengguna.

#### 4.4 Kebutuhan Non-Fungsional

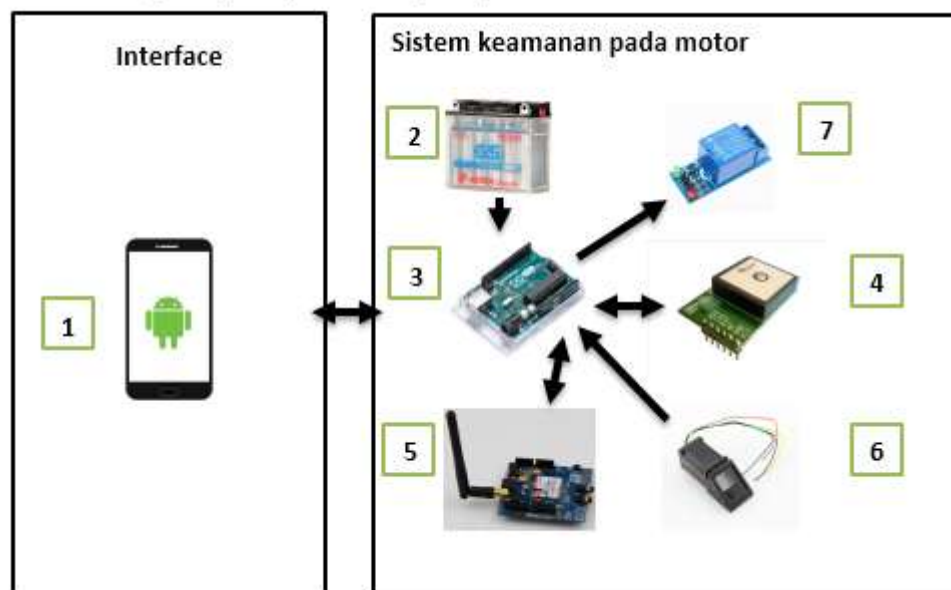
Pada kebutuhan non-fungsional menggunakan daya (aki motor atau adapter) 12v pada sistem. Sehingga sistem dapat bekerja sesuai dengan yang telah diinginkan. Pengunan juga dapat berinteraksi dengan sistem melalui LCD 16x2 yang dapat memonitorng hasil dari kerja sistem dan dapat berinteraksi langsung dengan cara memberi perintah melalui aplikasi pada *smartphone* yang telah terhubung dengan sistem, serta saat menggunakan pemindaian sidik jari.

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan dibahas mengenai konsep dasar, gambaran umum pada sistem, analisa kebutuhan sistem, diagram blok, perancangan perangkat keras dan perancangan perangkat lunak sistem.

### 5.1 Gambaran Umum Sistem

Dalam perancangan keras ini meliputi rangkaian mikrokontroler yang di dalamnya terdapat module *GPS* SKM53, module SIM900A, dan module *fingerprint*. Untuk lebih jelasnya dapat dilihat pada Gambar 5.1.



Gambar 5.1 Diagram Blok Sistem Keamanan Motor

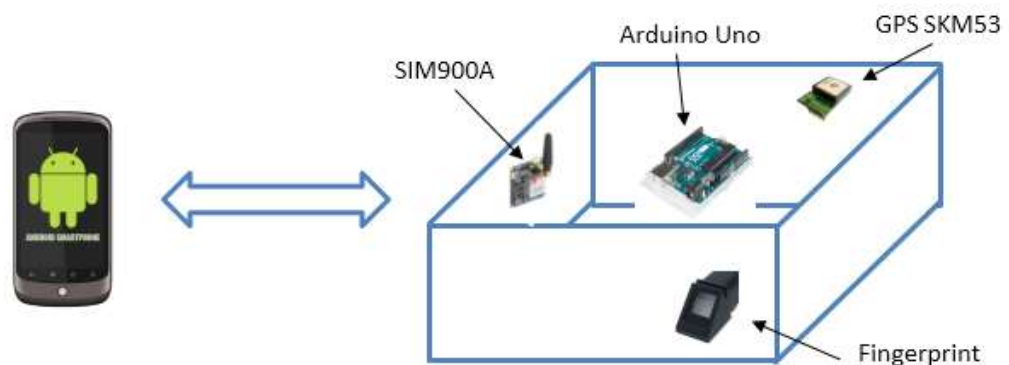
Keterangan:

1. *Smartphone*
2. Aki
3. Mikrokontroler arduino uno
4. Module *GPS* SKM53
5. Module SIM900A
6. Module *fingerprint*
7. Module relay

Pada gambar 5.1 tersebut dapat dilihat merupakan penjabaran dan penjelasan mengenai diagram blok perancangan sistem yang dilakukan pada interface dan sistem keamanan pada motor. Dalam sistem keamanan tersebut terdapat perancangan yang dimulai mikrokontroler, perancangan aki, serta perancangan proses keamanan yang telah dipasang pada kendaraan motor.

## 5.2 Perancangan Sistem

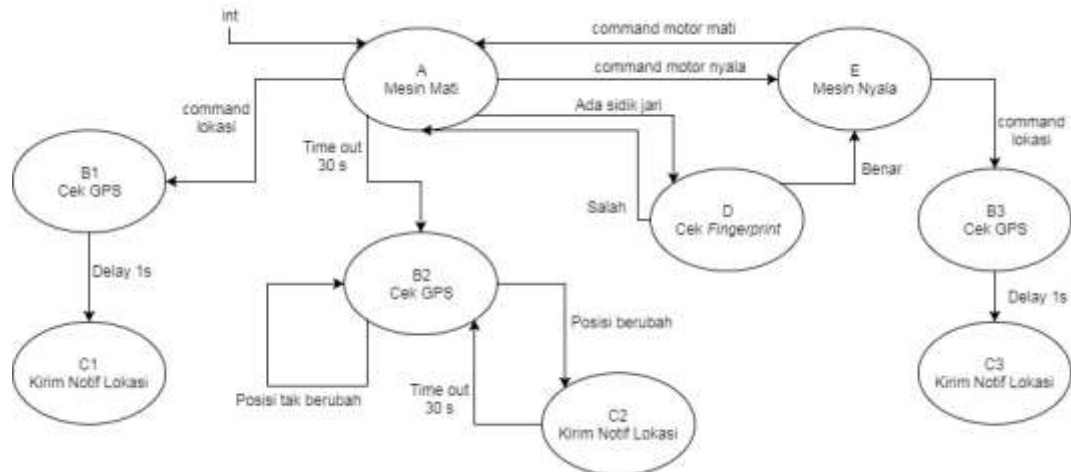
Pada bab rekayasa kebutuhan telah dijelaskan kebutuhan-kebutuhan yang harus dipenuhi untuk merancang sistem pada penelitian ini. Untuk itu perancangan sistem ini akan dilakukan perancangan *frame* sistem keamanan motor dan perancangan komunikasi port arduino. Perancangan perangkat keras berupa pembuatan *frame* sistem keamanan motor yang berisi modul sensor, SIM900A dan Arduino uno. Gambaran sederhana *frame* sistem keamanan motor beserta penempatan alat-alatnya dapat dilihat pada Gambar 5.2 perancangan sistem.



Gambar 5.2 Perancangan Sistem

### 5.2.1 Perancangan *State Machine* Pada Sistem Keamanan Motor

Pada perancangan *state machine* pada sistem keamanan motor ini akan menggunakan diagram blok agar dapat mempermudah dalam penerapannya pada sistem. Diagram blok *state machine* untuk sistem keamanan motor dibuat berdasarkan fungsi – fungsi dan tujuan dari sistem keamanan motor ini yang dapat dilihat pada Gambar 5.3 berikut.



**Gambar 5.3 Diagram Blok State Machine Pada Sistem Keamanan Motor**

Pada gambar diagram blok di atas menampilkan sebuah diagram blok dari *State machine* sistem keamanan motor. Untuk penjelasan dari diagram blok di atas dapat dilihat pada penjabaran di bawah ini.

*State A* merupakan *State* awal dari sistem keamanan motor ini. *State A* memiliki empat aksi untuk menjalankan *State – State* berikutnya. Untuk memanggil *State B1*, *State A* akan memberikan sebuah *command* terlebih dahulu untuk memanggil *State B1*, setelah *State B1* mendapatkan event dari *State A*, *State B1* akan memanggil *State C1* ketika delay telah mencapai 1 detik.

Kemudian untuk memanggil *State B2*, adanya timeout selama 30 detik dari *State A*. Setelah *B2* dijalankan akan menghasilkan dua aksi, yaitu posisi berubah atau posisi tidak berubah. Jika posisi berubah, secara otomatis akan masuk ke dalam *State C2* yang kemudian akan kembali ke *State B2* ketika timeout telah mencapai waktu 30 detik. Sedangkan, jika posisi tidak berubah maka aksi yang dilakukan hanya akan kembali ke *State B2*.

Berikutnya menjalankan *State D* dengan aksi ada sidik jari setelah dari *State A*. Pada *State D* ini memiliki dua aksi ketika dijalankan, yaitu benar dan salah. Jika aksi yang didapat adalah salah, maka akan kembali ke *State A*. Namun, jika benar maka akan menjalankan *State E*.

Untuk menjalankan *State E*, dibutuhkan dua aksi yaitu aksi dari *State D* yang sebelumnya sudah dijelaskan alurnya dan aksi dari *State A* yang merupakan aksi *command* untuk menjalankan *State E*. Pada *State E* ini juga terdapat dua aksi untuk menjalankan *State* berikutnya, untuk aksi pertama digunakan untuk kembali ke *State A* dengan aksi jika ada *command*. Sedangkan aksi berikutnya digunakan untuk memanggil *State B3* dengan aksi berupa *command*. Pada *State B3* secara otomatis akan menjalankan *State C3* ketika delay telah mencapai 1 detik.

### 5.2.2 Perancangan *Frame* Sistem Keamanan Motor

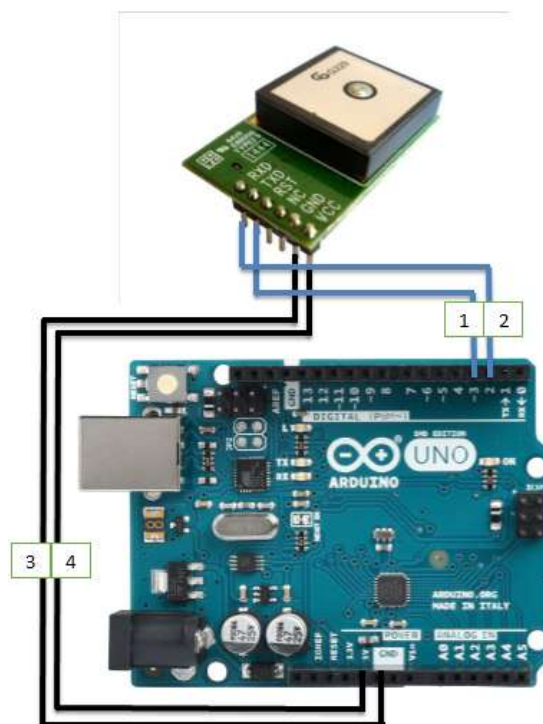
Perancangan *Frame* sistem keamanan motor berupa perancangan mekanik dari sistem. Dalam perancangan mekanik ini menggunakan papan akrilik sebagai dasar dari pembuatan alat sistem keamanan motor ini. Pada rancangan ini

semua tempat bagian tercantum seperti: mikrokontroler arduino uno, peletakan module-module, dan juga gambar akhir dari sistem keamanan motor yang digunakan.

Pada perancangan desain mekanik alat keamanan motor ini terdapat 4 module yang digunakan, meliputi:

#### **5.2.2.1 Perancangan Sistem Pembacaan Lokasi Pada Modul GPS SKM53**

Pada perancangan modul GPS SKM53 yang berfungsi sebagai pembaca lokasi terdapat beberapa komponen dan bagian-bagian pin yang terhubung pada port Arduino Uno. Untuk lebih jelas komponen yang di gunakan dan port yang terhubung pada Arduino Uno dapat di lihat pada Gambar 5.4.



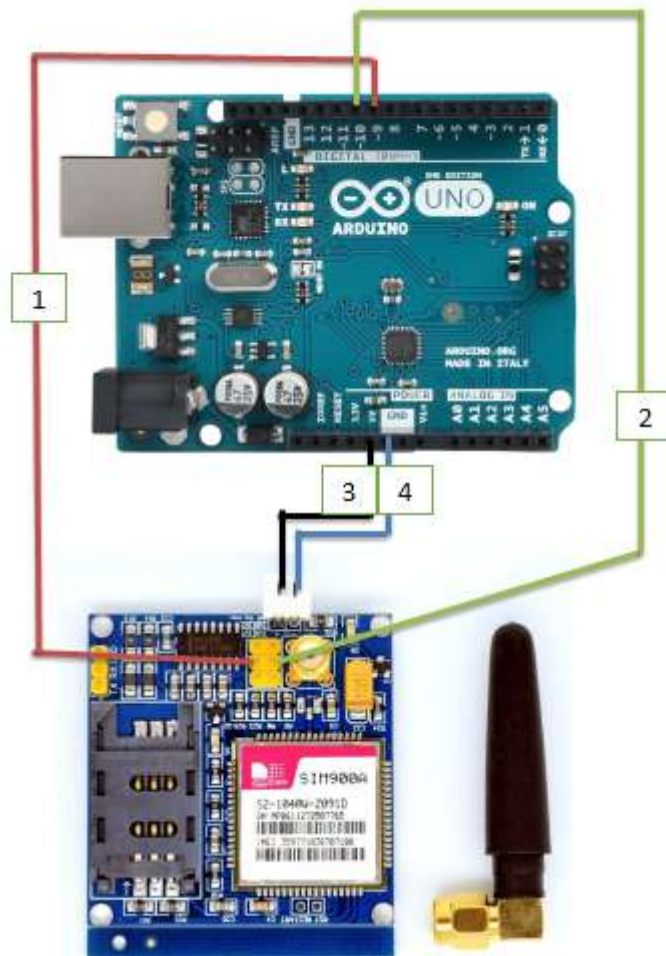
**Gambar 5.4 Perancangan Pembacaan Module GPS SKM53**

Pada Gambar 5.4 dapat dilihat pin Vcc terhubung dengan port 5 volt pada jalur dengan nomor 4 dan pin GDN terhubung dengan port GDN pada Arduino Uno dengan jalur nomor 3, sedangkan pada pin RXD SKM53 terhubung pada port 2 di arduino uno dengan nomor 2 dan pin TXD SKM53 terhubung pada port 3 di arduino uno dengan nomor 1.

#### **5.2.2.2 Perancangan Sistem Kerja Pada Modul SIM900A**

Pada perancangan modul SIM900A terdapat beberapa bagian pin-pin yang terhubung pada port Arduino Uno. Untuk lebih jelas dapat di lihat pada Gambar 5.5.



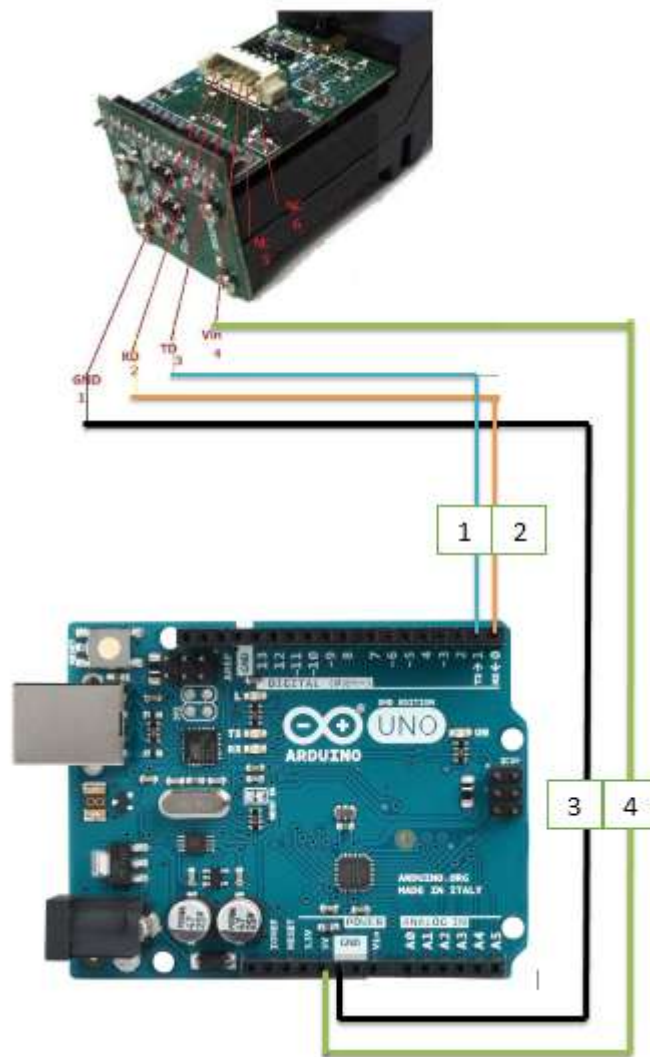


**Gambar 5.5 Perancangan Module SIM900A**

Pada Gambar 5.5 dapat dilihat terdapat 4 kabel *jumper* sebagai penghubung antara modul dengan mikrokontroler arduino uno, pin Vcc pada modul terhubung dengan port Vcc yang ada pada Arduino uno yang di jelaskan pada gambar terhubung dengan kabel nomor 3, sedangkan pin GDN pada modul terhubung dengan port GDN yang terdapat pada Arduino uno terhubung dengan kabel nomor 4. Pada kabel dengan nomor 2 yang menghubungkan modul menuju port 10 yang terdapat pada Arduino Uno, sedangkan pada kabel nomor 1 menghubungkan modul dengan port 9 pada arduino uno.

### **5.2.2.3 Perancangan Pembacaan Sensor *Fingerprint***

Pada perancangan pembacaan sensor *fingerprint* ini terhubung dengan pin - pin Arduino uno yang tersedia, untuk lebih jelas dapat di lihat pada Gambar 5.6.

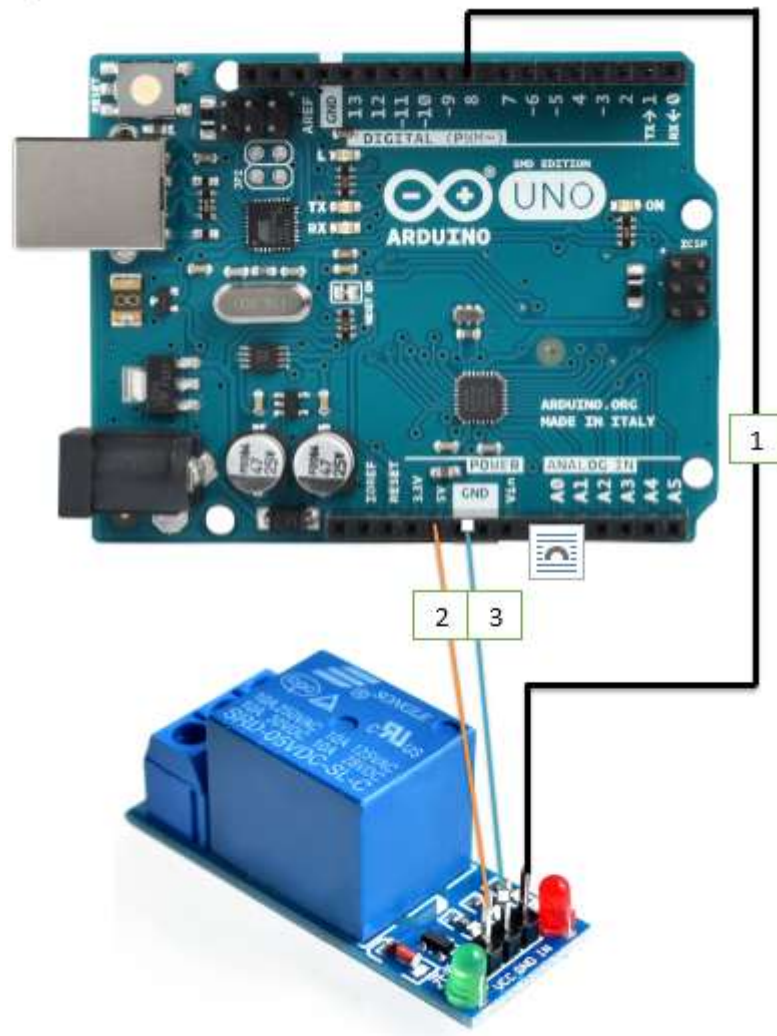


**Gambar 5.6 Perancangan Sensor *Fingerprint***

Pada Gambar 5.6 terdapat 4 kabel yang menghubungkan antara sensor *fingerprint* dan mikrokontroler arduino uno. Pada kabel dengan nomor 4 menghubungkan antara pin vcc di sensor ke port vcc pada arduino, sedangkan untuk kabel dengan nomor 3 menghubungkan antara pin gnd di sensor ke port gnd pada arduino. Untuk kabel dengan nomor 1 menghubungkan pin txd pada sensor ke port 1 pada arduino dan untuk nomor 2 menghubungkan pin rxd pada sensor ke port 0 pada arduino.

#### **5.2.2.4 Perancangan Modul Relay 1 Channel**

Pada perancangan modul relay 1 channel ini terhubung dengan pin -pin Arduino uno yang tersedia, untuk lebih jelas dapat di lihat pada Gambar 5.7.

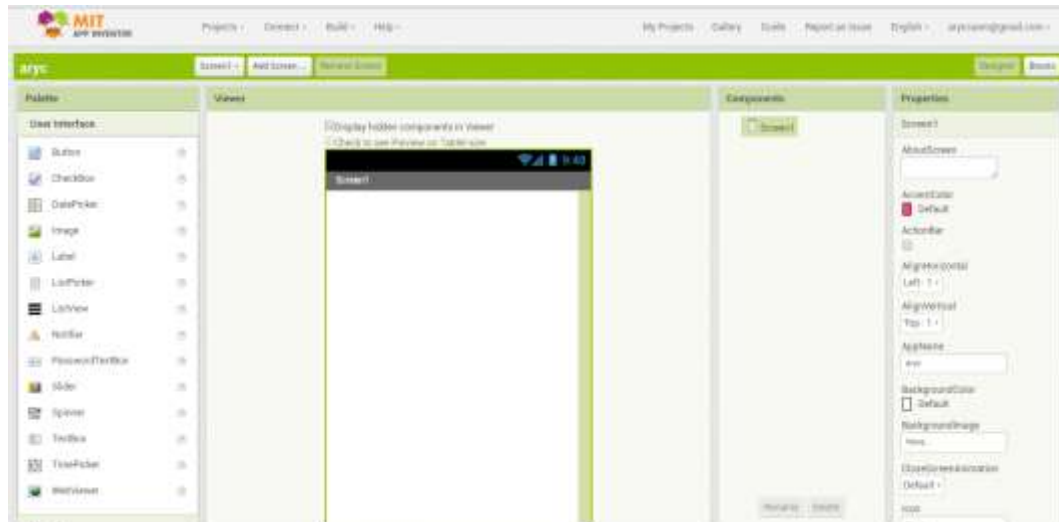


**Gambar 5.7 Perancangan Module Relay 1 Channel**

Pada Gambar 5.7 terdapat 3 kabel yang menghubungkan antara modul relay 1 channel dengan mikrokontroler arduino uno. Pada kabel dengan nomor 2 menghubungkan antara pin vcc di modul ke port vcc pada arduino, sedangkan untuk kabel nomor 3 menghubungkan antara pin gnd di modul ke port gnd pada arduino. Untuk nomor 1 menghubungkan pin IN pada modul ke port 8 pada arduino.

### **5.2.3 Perancangan Aplikasi Menggunakan AppInverter**

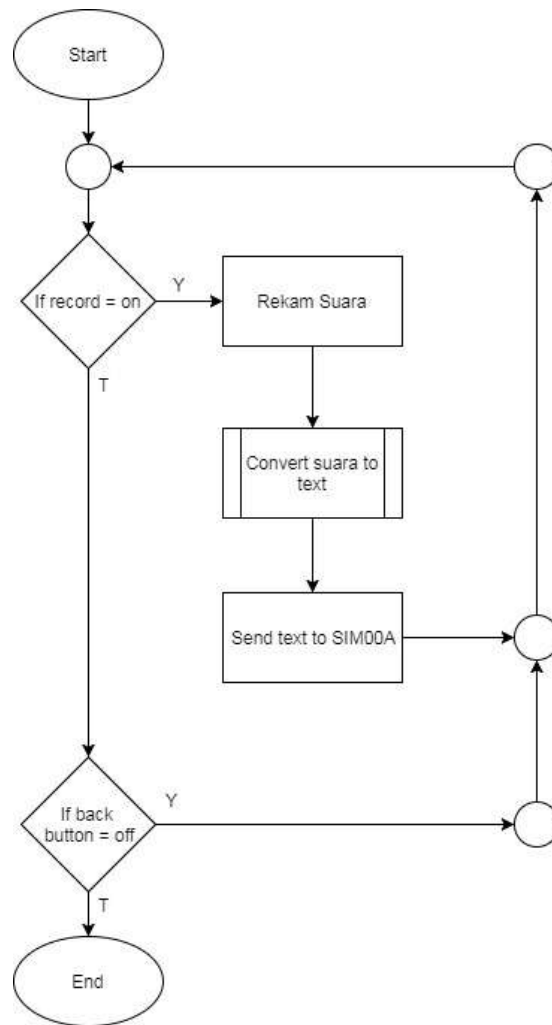
Berikut ini merupakan tampilan dari interface yang ada pada software *App inverter*:



**Gambar 5.8 Tampilan Awal App Inverter**

Pada Gambar 5.8 di atas merupakan tampilan awal pada *App inverter*, dapat dilihat pada Gambar 5.8 terdapat sebuah tempat yang digunakan sebagai tempat untuk medesain tampilan yang kemudian akan digunakan pada *smartphone*. Pada Gambar 5.8 tepatnya di sebelah kanan atas terdapat sebuah *button blocks* yang jika digunakan dapat mengantarkan *user* ke bagian *coding block diagram*.

Untuk mempermudah penulis dalam perancangan dan pembuatan aplikasi dengan menggunakan *appinverter* ini, maka penulis menggambarkan dalam bentuk *flowchart*, sehingga hasil dari perancangan ini dapat berjalan sesuai dengan yang diharapkan. Untuk *flowchart* sistem kerja pada aplikasi dapat dilihat pada Gambar 5.9.



**Gambar 5.9 Flowchart Perancangan Aplikasi Pada Smartphone**

Dengan adanya *flowchart* tersebut dapat membantu perancangan dan pembuatan aplikasi untuk *smartphone* agar fungsi yang diharapkan dapat berjalan dengan baik.

### 5.3 Implementasi Sistem

Implementasi sistem ini meliputi implementasi pada perangkat keras dan perangkat lunak yang di gunakan.

#### 5.3.1 Implementasi State Machine Pada Sistem Keamanan Motor

Implementasi *state machine* dalam program arduino uno ini penulis melakukan penelitian yang di bantu beberapa pihak terkait menggunakan fungsi dari *state machine* untuk mempermudah proses pembuatan sistem ini. Untuk perancangan dapat di lihat pada tabel 5.1

**Table 5.1 Potongan Program *State Machine***

1	////////////////////Inisialisasi Machine////////////////	<i>State</i>
2	<i>State State_A</i> (&Motor_OFF, NULL, NULL);	
3	<i>State State_B1</i> (&skm53, NULL, NULL);	
4	<i>State State_C1</i> (&SendMessage, NULL, NULL);	
5	<i>State State_B2</i> (&skm53, NULL, NULL);	
6	<i>State State_C2</i> (&SendMessage, NULL, NULL);	
7	<i>State State_D</i> (&getFingerprintIDez, NULL, NULL);	
8	<i>State State_E</i> (&Motor_ON, NULL, NULL);	
9	<i>State State_B3</i> (&skm53, NULL, NULL);	
10	<i>State State_C3</i> (&SendMessage, NULL, NULL);	
11		
12	Fsm fsm_A1 (&State_A);	
13	Fsm fsm_A2 (&State_A);	
14	Fsm fsm_A3 (&State_A);	
15	Fsm fsm_A4 (&State_A);	
16	Fsm fsm_B1 (&State_B1);	
17	Fsm fsm_B2 (&State_B2);	
18	Fsm fsm_C2 (&State_C2);	
19	Fsm fsm_D1 (&State_D);	
20	Fsm fsm_D2 (&State_D);	
21	Fsm fsm_E1 (&State_E);	
22	Fsm fsm_E2 (&State_E);	
23	Fsm fsm_B3 (&State_B3);	
24	void setup()	
25	{	
26	Serial.begin(9600);	
27	lcd.begin(16,4);	
28	pinMode(8,OUTPUT);	
29	fsm_A1.add_transition(&State_A, &State_B1, NULL, NULL);	

30	fsm_A2.add_timed_transition(&State_A, &State_B2, 30000, NULL);
31	fsm_A3.add_transition(&State_A, &State_D, NULL, NULL);
32	fsm_A4.add_transition(&State_A, &State_E, NULL, NULL);
33	fsm_B1.add_timed_transition(&State_B1, &State_C1, 1000, NULL);
34	fsm_B2.add_timed_transition(&State_B2, &State_C2, 1000, NULL);
35	fsm_C2.add_timed_transition(&State_C2, &State_B2, 30000, NULL);
36	fsm_D1.add_transition(&State_D, &State_E, NULL, NULL);
37	fsm_D2.add_transition(&State_D, &State_A, NULL, NULL);
38	fsm_E1.add_timed_transition(&State_E, &State_B3, 1000, NULL);
39	fsm_E2.add_timed_transition(&State_E, &State_A, 1000, NULL);
40	fsm_B3.add_timed_transition(&State_B3, &State_C3, 1000, NULL);
41	//////////////////////////////////// /
42	void StateMachine() {
43	getFingerprintIDez();
44	RecieveMessage();
45	skm53();
46	fsm_A2.run_machine();
47	if(strTemp!=Lat_sebelumnya  strTemp1!=Lon_sebelumnya){fsm_B2.run_machine();fsm_C2.run_machine();}
48	if(finger.fingerID==id){kondisi=1;}
49	if(SIM900A.available()=='A'){fsm_A1.run_machine();fsm_B1.run_machine();}
50	if(SIM900A.available()=='B'){fsm_A4.run_machine();kondisi=1;}
51	if(kondisi==0){fsm_D2.run_machine();}
52	while(kondisi==1){
53	fsm_A3.run_machine();
54	fsm_D1.run_machine();
55	RecieveMessage();

56	<code>if (SIM900A.available()=='A') {</code>
57	<code>  fsm_E1.run_machine();</code>
58	<code>  fsm_B3.run_machine();</code>
59	<code>}</code>
60	<code>if (SIM900A.available()=='C') {</code>
61	<code>  fsm_E2.run_machine();</code>
62	<code>  kondisi=0;</code>
63	<code>  return;</code>
64	<code>}</code>
65	<code>}</code>
66	
67	<code>  Lat_sebelumnya=strTemp;</code>
68	<code>  Lon_sebelumnya=strTemp1;</code>
69	
70	<code>}</code>

Pada baris 2 hingga baris 10 merupakan inisialisasi *state machine* dengan program yang akan dijalankan. Kemudian pada baris 12 hingga baris 23 merupakan inisialisasi setiap *state* menjadi fungsi *finite state machine* atau dapat disingkat *FSM*. Pada baris 29 hingga baris 40 merupakan pengaturan untuk *event* dari perubahan atau perpindahan setiap *state* yang ada, yang dijelaskan setiap barisnya sebagai berikut:

Baris ke 29 merupakan *event transition state A ke state B1*.

Baris ke 30 merupakan *event transition state A ke state B2* dengan tambahan waktu *delay* selama 30000ms.

Baris ke 31 merupakan *event transition state A ke state D*.

Baris ke 32 merupakan *event transition state A ke state E*.

Baris ke 33 merupakan *event transition state B1 ke state C1* dengan tambahan waktu *delay* selama 1000ms.

Baris ke 34 merupakan *event transition state B2 ke state C2* dengan tambahan waktu *delay* selama 1000ms.

Baris ke 35 merupakan *event transition state C2 ke state B2* dengan tambahan waktu *delay* selama 30000ms.

Baris ke 36 merupakan *event transition state D ke state E*.

Baris ke 37 merupakan *event transition state D ke state A*.



Baris ke 38 merupakan *event transition state* E ke *state* B3 dengan tambahan waktu *delay* selama 1000ms.

Baris ke 39 merupakan *event transition state* E ke *state* A dengan tambahan waktu *delay* selama 1000ms.

Baris ke 40 merupakan *event transition state* B3 ke *state* C3 dengan tambahan waktu *delay* selama 1000ms.

Pada baris 42 merupakan fungsi untuk menjalankan program utama. Pada baris 46 hingga baris 70 merupakan program utama dari sistem yang memanfaatkan fungsi dari *state machine* yang dijelaskan setiap barisnya sebagai berikut:

Baris ke 46 merupakan fungsi untuk menjalankan *transisi state* A ke *state* B2.

Baris ke 47 merupakan fungsi untuk membandingkan data longitude dan latitude, serta menjalankan fungsi untuk melakukan *transisi state* B2 ke *state* C2 dan *transisi state* C2 ke *state* B2.

Baris ke 49 merupakan fungsi pembacaan pesan yang diterima oleh modul SIM900A dengan pesan "A", serta menjalankan fungsi untuk melakukan *transisi state* A ke *state* B1 dan *transisi state* B1 ke *state* C1.

Baris ke 50 merupakan fungsi pembacaan pesan yang diterima oleh modul SIM900A dengan pesan "B", serta menjalankan fungsi untuk melakukan *transisi state* A ke *state* E.

Baris ke 51 merupakan fungsi untuk menjalankan *transisi state* D ke *state* A.

Baris ke 53 merupakan fungsi untuk menjalankan *transisi state* A ke *state* D.

Baris ke 54 merupakan fungsi untuk menjalankan *transisi state* D ke *state* E.

Baris ke 57 merupakan fungsi untuk menjalankan *transisi state* E ke *state* B3.

Baris ke 58 merupakan fungsi untuk menjalankan *transisi state* B3 ke *state* C3.

Baris ke 61 merupakan fungsi untuk menjalankan *transisi state* E ke *state* A.

### 5.3.2 Implementasi Frame Sistem Keamanan Motor

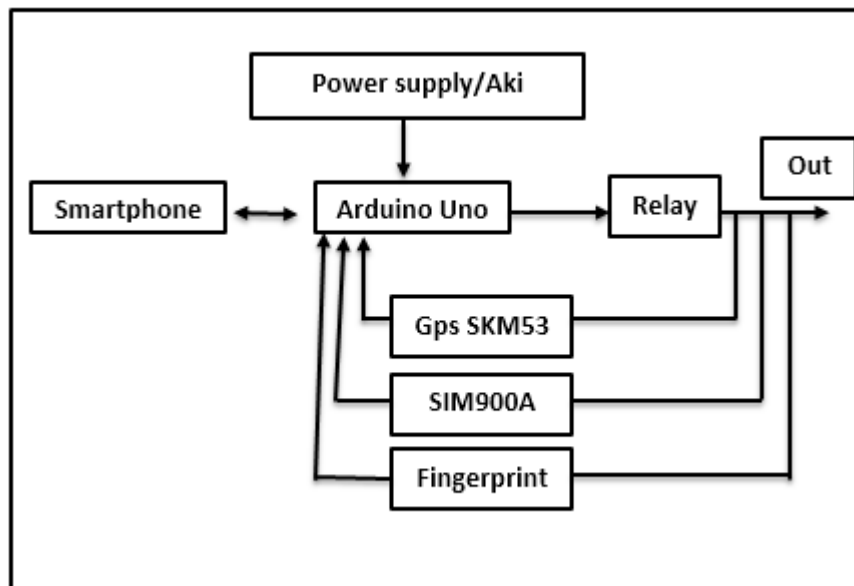
Implementasi perangkat keras sistem keamanan pada motor yang digunakan yaitu, mikrokontroler arduino uno. Module-module yang digunakan adalah module *GPS* SKM53, module SIM900A, dan module *fingerprint*. Tegangan yang dibutuhkan sistem ini untuk bekerja sebesar 12 V. Pada implementasi perangkat keras ini juga akan dijelaskan dan dijabarkan mengenai spesifikasi pembuatan yang ada pada sistem keamanan kendaraan motor ini yang meliputi:

1. Penentuan modul elektronik yang akan digunakan, sebagai berikut:

- a. *Smartphone* dengan o.s. android
- b. Power supply / aki

- c. Arduino Uno
- d. Module *GPS* SKM53
- e. Module SIM900A
- f. Sensor *fingerprint*
- g. Module relay 1 channel

## 2. Skema pembuatan perangkat keras



**Gambar 5.10 Skema Pembuatan Perangkat Keras**

Prinsip kerja pada sistem ini dijabarkan sebagai berikut:

- Catu daya dengan tegangan sebesar 12v sebagai sumber daya rangkaian untuk menjalankan sistem berasal dari power supply/aki
- Catu daya arduino uno juga diambil dari power supply/aki, karena sumber daya tersebut langsung dihubungkan pada papan arduino uno.
- Catu daya sebesar 5v dari mikrokontroler digunakan untuk mencatu module *GPS* SKM53, dan mikrokontroler arduino uno.
- Catu daya sebesar 5v dari mikrokontroler digunakan untuk mencatu module SIM900A, dan mikrokontroler arduino uno.
- Catu daya sebesar 5v dari mikrokontroler digunakan untuk mencatu sensor *fingerprint*, dan mikrokontroler arduino uno
- SMS yang dikirim melalui *smartphone* dan masuk ke SIM900A kemudian diterima oleh arduino uno yang kemudian digunakan untuk

dapat mengeksekusi proses nyala dan mati motor, serta mencari lokasi melalui *GPS*.

Implementasi pada sistem keamanan motor ini menggunakan Arduino Uno, module *GPS* SKM53 sebagai pembaca lokasi dari motor yang akan mengirimkan data untuk pengguna, module SIM900A sebagai komunikasi pada sistem, module relay 1 channel sebagai kontrol hidup dan nyala pada *starter* motor, dan sensor *fingerprint* sebagai pendeteksi kepastian pemilik motor. Hasil pembacaan data dari sensor dapat dilihat melalui LCD 16X4 untuk mengetahui bahwa sidik jari pengguna cocok atau tidak dan bahwa motor dalam kondisi nyala atau mati, sistem ini menggunakan tenaga 12V yang di mana tegangan 12V di tujukan untuk daya yang di butuhkan pada arduino uno dan sistem, agar sistem yang ada bekerja sesuai dengan yang di inginkan. Untul lebih jelasnya dapat di lihat pada Gambar 5.11.



**Gambar 5.11 Bentuk Fisik Sistem Keamanan Motor**

Pada bentuk fisik sistem keamanan motor menggunakan papan akrilik sebagai bahan dasar pembuatan wadah alat.



**Gambar 5.12 Implementasi Sistem Keamanan Motor**

Pada Gambar 5.12 dapat dilihat bahwa alat sistem keamanan motor terbuat dari bahan akrilik. Pada penerapannya alat sistem keamanan motor ini diterapkan di atas *tank* bensin motor dan kabel *power supply* yang terhubung ke aki.

#### **5.3.2.1 Implementasi Sistem Pembacaan Lokasi Pada Modul GPS SKM53**

Implementasi module *GPS* ini penulis menggunakan module *GPS SKM53* yang dihubungkan dengan mikrokontroler arduino menggunakan 4 kabel *jumper female-female* agar dapat melakukan proses pembacaan lokasi. Data – data lokasi yang didapat dari *GPS* akan diproses pada arduino agar dapat dibandingkan lokasi awal dengan lokasi sekarang. Untuk gambar dari implementasi sensor ini dapat dilihat pada Gambar 5.13.



**Gambar 5.13 Implementasi GPS SKM53**

#### **5.3.2.2 Implementasi Sistem Kerja Pada Modul SIM900A**

Implementasi module GSM/GPRS ini penulis menggunakan module SIM900A yang bertujuan untuk sebagai komunikasi tunggal pada sistem keamanan motor ini, dengan menggunakan SIM900A ini perintah-perintah yang diberikan pengguna dari aplikasi pada *smartphone* akan dikirim melalui via sms ke SIM900A dan ketika diterima oleh SIM900A pesan akan diproses pada arduino uno, dengan SIM900A juga dapat mengirimkan data lokasi dari arduino kepada pengguna. Untuk gambaran module SIM900A ini dapat di lihat pada Gambar 5.14.



**Gambar 5.14 Implementasi SIM900A**

### **5.3.2.3 Implementasi Pembacaan Sensor *Fingerprint***

Implementasi sensor *fingerprint* ini penulis menggunakan sensor *fingerprint* yang ada di pasaran yang bertujuan untuk membaca sidik jari dari pengguna, yang mana apabila sidik jari yang dibaca sesuai dengan sidik jari pengguna maka sensor akan mengirimkan nilai pada relay untuk menyalakan motor, sedangkan apabila sidik jari yang dibaca tidak sesuai dengan sidik jari pengguna maka sensor akan mengirimkan nilai pada relay agar relay tetap dalam kondisi mati. Untuk gambaran dari sensor *fingerprint* ini dapat dilihat pada Gambar 5.15.



**Gambar 5.15 Implementasi Sensor *fingerprint***

#### **5.3.2.4 Implementasi Modul Relay 1 Channel**

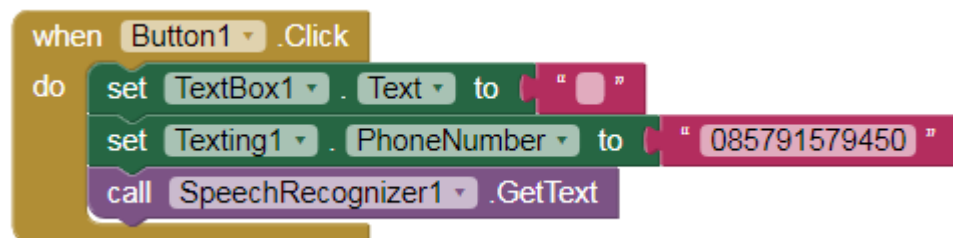
Implementasi module relay 1 channel bertujuan sebagai kontrol atau *switch* pada *starter* motor, jika relay bernilai 1 maka *starter* dalam kondisi yang bisa dinyalakan, sedangkan sebaliknya jika nilai relay 0 maka *starter* dalam kondisi yang tidak bisa dinyalakan. Untuk gambaran module relay 1 channel ini dapat dilihat pada Gambar 5.16.



**Gambar 5.16 Implementasi Relay 1 Channel**

### 5.3.3 Implementasi Aplikasi Menggunakan *AppInverter*

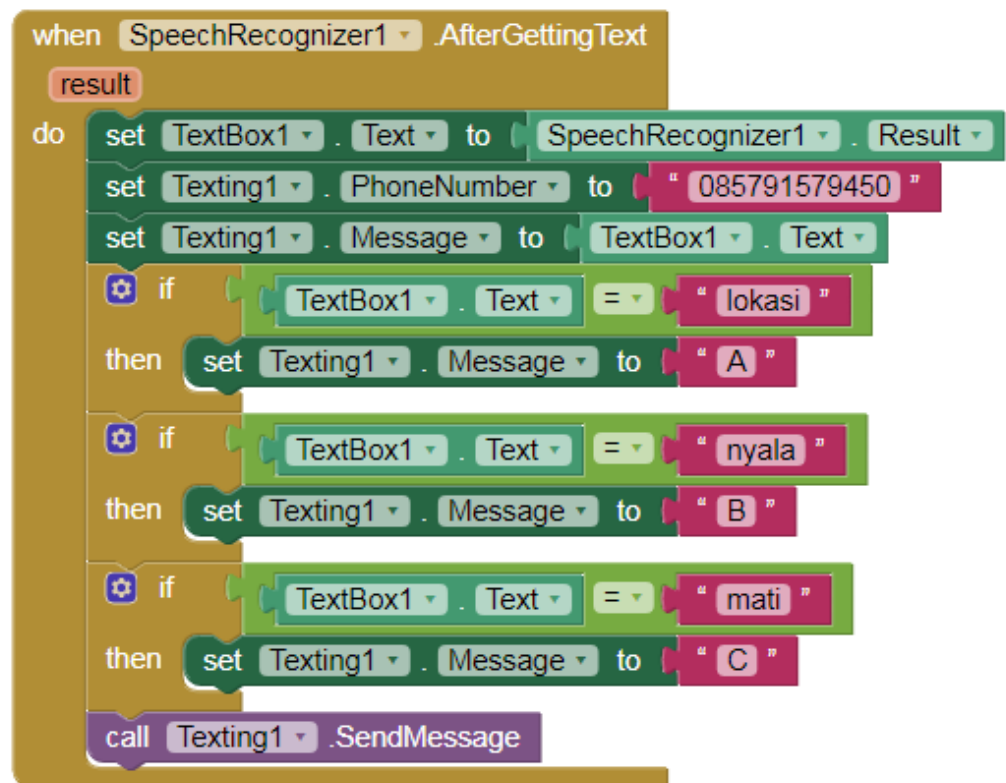
Dalam pengimplementasian aplikasi dengan menggunakan *appinverter* ini akan mengikuti dasar perancangan yang telah dijabarkan sebelumnya.



**Gambar 5.17 Block Coding Speech Recognizer**

Pada Gambar 5.17 merupakan blok diagram yang berguna untuk menjalankan proses inialisasi *textbox*, *phone number*, dan merekam suara dari user. Saat program jalan, program akan secara otomatis membuat sebuah kotak teks yang akan diisi oleh inputan dari pengguna dan melakukan *setting* pada nomor handphone dengan memasukkan nomor yang berada pada module SIM900A. Setelah kedua proses di atas telah dijalankan, kemudian program akan memulai merekam suara, menunggu masukkan suara dari pengguna.





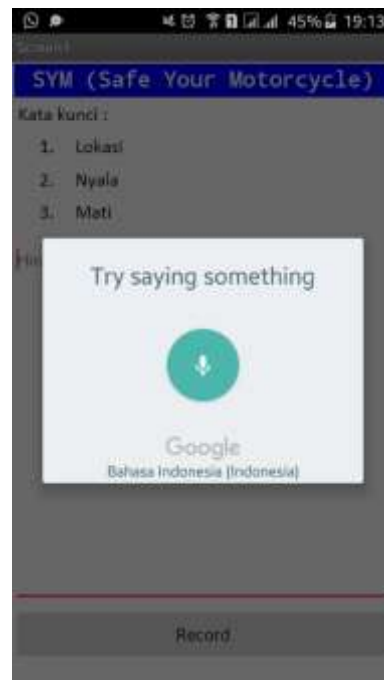
**Gambar 5.18 Block Coding Text and Sending Message Process**

Pada Gambar 5.18 merupakan blok diagram yang berguna untuk menjalankan proses *convert speech to text*, fungsi *IF-ELSE* dan mengirim sms. Program secara otomatis jalan ketika pengguna telah memberikan masukan pada aplikasi. Saat program berjalan, program akan melakukan *convert* suara menjadi teks. Kemudian suara yang telah diubah menjadi teks dimasukkan dalam fungsi *IF-ELSE* agar ketika dikirimkan ke sistem pada motor menjadi lebih mudah untuk dibaca sistem pada motor. Setelah teks selesai melakukan fungsi *IF-ELSE*, teks akan dikirim melalui sms ke nomor pada SIM900A.



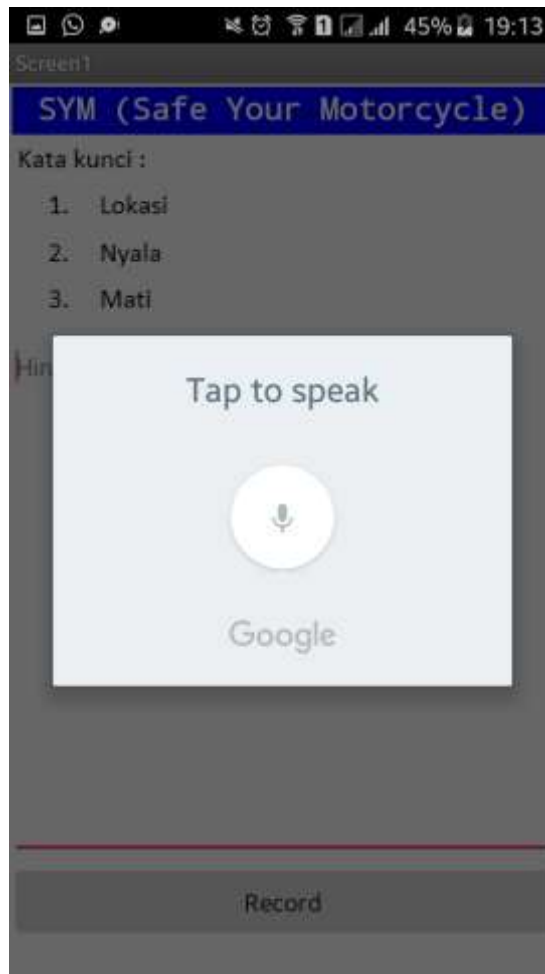
**Gambar 5.19 Tampilan Aplikasi pada *Smartphone***

Pada Gambar 5.19 dapat dilihat terdapat tombol *record*, tombol *record* ini berfungsi untuk menejalankan aplikasi dan memulai proses untuk merekam suara dari pengguna.



**Gambar 5.20 Tampilan Aplikasi Ketika Menunggu Input Suara dari Pengguna**

Pada Gambar 5.20 menunjukan aplikasi pada *smartphone* setelah pengguna menekan tombol *record*. Seperti pada gambar di atas, aplikasi menunggu inputan suara dari user untuk melakukan eksekusi berikutnya.



**Gambar 5.21 Tampilan Aplikasi Ketika Pengguna Telah Meng-inputkan Suara**

Pada Gambar 5.21 menunjukkan aplikasi telah menerima atau merekam suara dari pengguna. Setelah suara berhasil direkam oleh aplikasi, langkah berikutnya aplikasi akan menampilkan hasil data dari rekaman dalam bentuk teks pada kotak yang tertera pada aplikasi dan kemudian secara otomatis mengirimkan pesan pada sistem keamanan motor, seperti yang ditunjukkan pada Gambar 5.23.



**Gambar 5.22 Tampilan Aplikasi Ketika Data Ditampilkan dan Dikirim**

Pada Gambar 5.22 menunjukkan bahwa aplikasi sedang menampilkan data yang sebelumnya berasal dari suara pengguna diubah menjadi data dalam bentuk teks. Kemudian data tersebut dikirimkan kepada sistem keamanan motor melalui pesan singkat yang ditunjukkan dengan tulisan "*message sent*".

## BAB 6 PENGUJIAN DAN ANALISIS

### 6.1 Pengujian *State* Sistem Keamanan Motor

Pada pengujian *State* sistem keamanan motor ini akan diuji apakah alat atau sistem keamanan motor ini dapat bekerja dengan baik. Agar dalam pengujian dapat berjalan dengan baik, oleh karena itu diperlukan adanya strategi uji dan skenario dalam pengujian tersebut. Strategi dan skenario pengujian pada sistem akan dilakukan sesuai dengan banyaknya aksi atau kejadian ketika perpindahan antara *state*. Setiap aksi perpindahan *state* akan diuji atau diambil data sebanyak 10 kali. Skenario pada pengujian setiap aksi perpindahan *state* akan disesuaikan dengan aksi yang terjadi. Dengan adanya skenario dan strategi uji ini dapat mempermudah dalam melakukan pengujian.

Selain membutuhkan skenario dan strategi uji sebelum melakukan pengujian juga diperlukan prosedur dari pengujian yang dapat dilihat pada tabel 6.1.

**Table 6.1** Prosedur Pengujian Sistem Keamanan Motor

Kasus Pengujian	Mengambil data dari setiap <i>State</i> yang telah diproses dan dijalankan oleh arduino uno dan pengguna pada <i>smartphone</i> .
Objek Pengujian	Alat sistem keamanan motor yang terbuat dari papan akrilik dan <i>smartphone</i> .
Tujuan Pengujian	Untuk menguji hasil dari setiap <i>State</i> yang telah diproses dan dijalankan oleh arduino uno dan pengguna pada <i>smartphone</i> .
Prosedur Pengujian	<ol style="list-style-type: none"><li>1. Menyiapkan alat sistem keamanan motor dan <i>smartphone</i> yang akan digunakan, serta <i>stopwatch</i> sebagai pencatat waktu.</li><li>2. Menempatkan sensor dan modul pada alat sistem keamanan motor.</li><li>3. Menghubungkan sensor <i>fingerprint</i>, <i>GPS</i>, relay dan SIM900A pada arduino uno.</li><li>4. Meng-<i>upload</i> program pada arduino uno.</li><li>5. Menjalankan sistem dan menguji setiap <i>State</i> yang diproses sistem.</li></ol>

#### 6.1.2 Pengujian *State* A : Mesin mati

Pada pengujian *State* A : Mesin mati akan dilakukan empat pengujian berdasarkan empat aksi yang dilakukan oleh *State* A, yaitu *command* memanggil lokasi, time out 30 s, pemindaian sidik jari dan *command* untuk menyalakan motor. Hasil dari pengujian ini akan ditampilkan dalam bentuk tabel dan pengujian ini diuji sebanyak 10 kali setiap aksi.

#### 6.1.2.1 Tujuan

Tujuan dari pengujian *state A* ini untuk mendapatkan hasil nilai uji berdasarkan aksi – aksi yang dilakukan, sehingga mampu dianalisis data dari hasil nilai uji.

#### 6.1.2.2 Prosedur

1. Pengujian aksi *command* memanggil lokasi

Kondisi motor dalam keadaan mati dan berada pada lokasi yang tetap atau sama, yaitu garasi. Kemudian dilakukan pengujian dengan cara memberikan *command* memanggil lokasi.

2. Pengujian aksi time out 30s

Kondisi motor dalam keadaan mati dan berpindah lokasi. Waktu pengujian dimulai pengujian pada jam 03.24 pagi dan setiap 30 detik yang telah dihitung pada *stopwatch* motor akan berhenti sampai data lokasi diterima, kemudian ketika data telah diterima akan dilakukan pengujian kembali.

3. Pengujian aksi pemindaian sidik jari

Sidik jari yang diuji coba menggunakan *sample* dari 4 orang atau 4 sidik jari. Dengan pengujian 2 orang melakukan 3 kali pengujian dan 2 melakukan 2 kali pengujian.

4. Pengujian aksi *command* untuk menyalakan motor

Kondisi awal motor dalam keadaan mati, kemudian diberikan *command* untuk menyalakan motor. Setelah motor dalam keadaan nyala, berikutnya dikembalikan ke kondisi keadaan mati lagi.

#### 6.1.2.3 Hasil

1. Pengujian aksi *command* memanggil lokasi

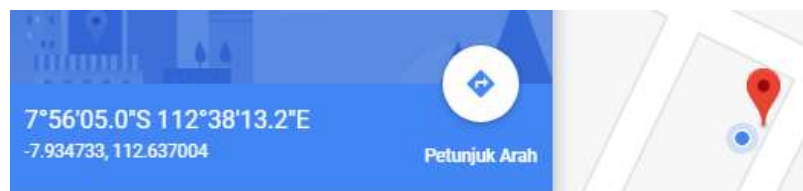
**Table 6.2 Hasil Pengujian Aksi *Command* Memanggil Lokasi**

Pengujian	Hasil
1	Mengirim ke pengguna data lokasi -7.934756, 112.636960
2	Mengirim ke pengguna data lokasi -7.934756, 112.636960
3	Mengirim ke pengguna data lokasi -7.934756, 112.636960
4	Mengirim ke pengguna data lokasi -7.934733, 112.637004*
5	Mengirim ke pengguna data lokasi -7.934756, 112.636960
6	Mengirim ke pengguna data lokasi -7.934756, 112.636960
7	Mengirim ke pengguna data lokasi -7.934756, 112.637005*
8	Mengirim ke pengguna data lokasi -7.934756, 112.636960

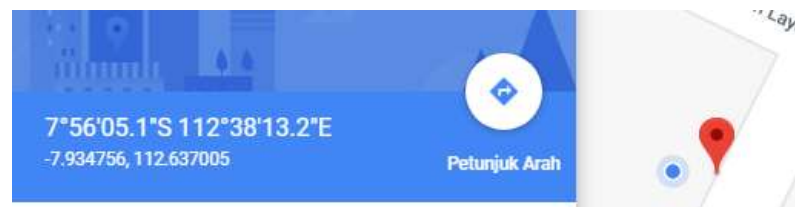
Pengujian	Hasil
9	Mengirim ke pengguna data lokasi -7.934756, 112.636960
10	Mengirim ke pengguna data lokasi -7.934756, 112.636960



**Gambar 6.1 Lokasi yang Benar**



**Gambar 6.2 Lokasi Pengujian ke – 4**



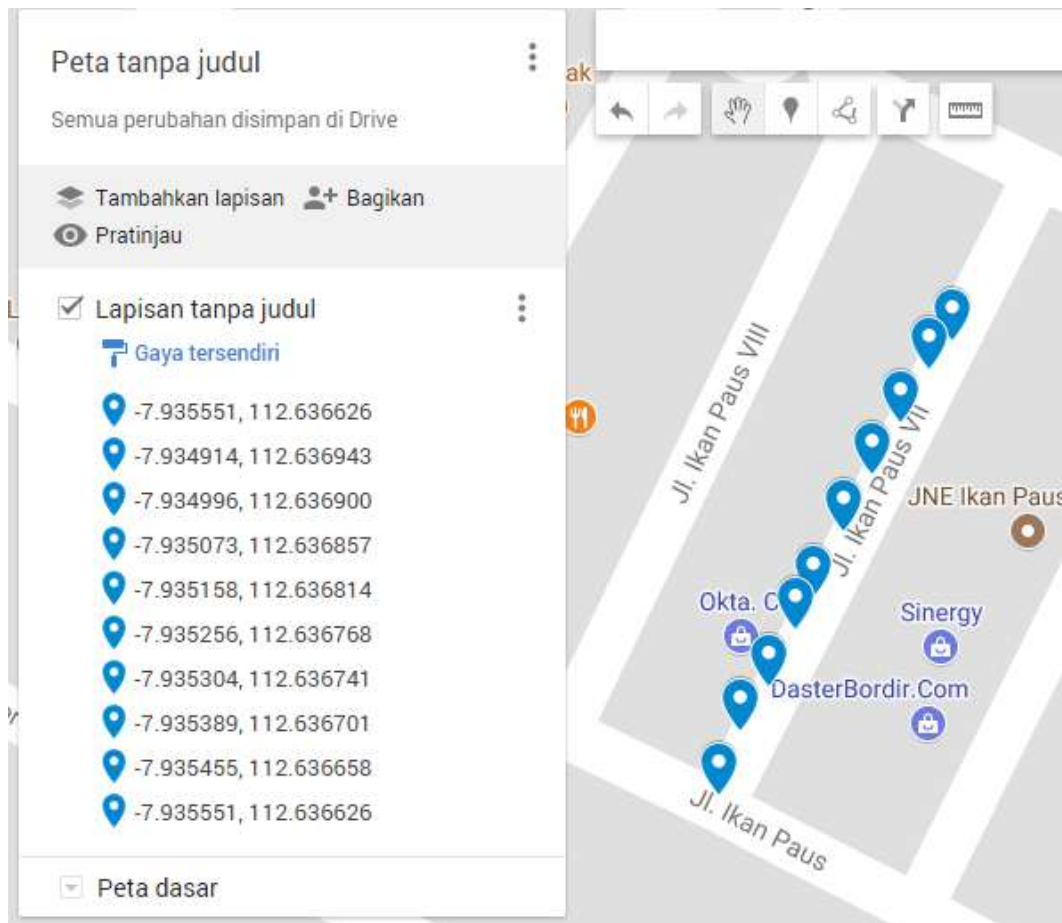
**Gambar 6.3 Lokasi Pengujian ke – 7**

## 2. Pengujian aksi time out 30s

**Table 6.3 Hasil Pengujian Aksi time out 30 s**

Pengujian	Selisih waktu(s)	Waktu data diterima(jam)	Data lokasi yang diterima
1	38.75	03.24	-7.934872, 112.636978
2	36.36	03.25	-7.934914, 112.636943
3	35.70	03.25	-7.934996, 112.636900
4	39.05	03.26	-7.935073, 112.636857
5	36.13	03.27	-7.935158, 112.636814
6	36.73	03.27	-7.935256, 112.636768

Pengujian	Selisih waktu(s)	Waktu data diterima(jam)	Data lokasi yang diterima
7	36.67	03.28	-7.935304, 112.636741
8	36.23	03.28	-7.935389, 112.636701
9	35.90	03.29	-7.935455, 112.636658
10	34.88	03.30	-7.935551, 112.636626



**Gambar 6.4 Koordinat dari pengujian State A ke State B2**

### 3. Pengujian aksi pemindaian sidik jari

**Table 6.4 Hasil Pengujian Pemindaian Sidik Jari**

Pengujian	Hasil
1	Terbaca
2	Terbaca
3	Terbaca
4	Terbaca



Pengujian	Hasil
5	Terbaca
6	Tidak Terbaca
7	Terbaca
8	Terbaca
9	Terbaca
10	Terbaca

4. Pengujian aksi *command* untuk menyalakan motor

**Table 6.5 Hasil Pengujian Aksi *Command* untuk Menyalakan Motor**

Pengujian	Hasil
1	Motor Nyala
2	Motor Nyala
3	Motor Nyala
4	Motor Nyala
5	Motor Nyala
6	Motor Nyala
7	Motor Nyala
8	Motor Nyala
9	Motor Nyala
10	Motor Nyala

#### **6.1.2.4 Analisis**

1. Pengujian aksi *command* memanggil lokasi

Pada Tabel 6.2 dapat dilihat terdapat dua data lokasi yang tidak sesuai pada pengujian ke 4 dan ke 7. Data *GPS* yang diterima ini terkadang mengalami perubahan dikarenakan faktor dari luar, seperti contohnya adalah kondisi cuaca dan tempat pembacaan *GPS* yang berada dalam rumah. Namun, dari data yang diuji lokasi yang berubah tidak jauh dari lokasi sebenarnya. Sehingga pada pengujian ini didapatkan hasil kesuksesan sistem sebesar 80%.

2. Pengujian aksi time out 30s

Pada Tabel 6.3 dapat dilihat bahwa setiap 30s aksi untuk *State B2* yaitu cek *GPS* berjalan karena pengguna menerima data berupa lokasi. Hasil dari pengujian ini yang ditunjukkan pada Tabel 6.3 rata – rata selisih waktu dari waktu time out 30s hingga diterima adalah 6.64 detik. Pada pengujian ini didapatkan hasil

kesuksesan sistem sebesar 100% dikarenakan data lokasi pada 10 kali pengujian telah diterima oleh pengguna. Berikut ini gambar dari koordinat lokasi.

### 3. Pengujian aksi pemindaian sidik jari

Hasil pada tabel di atas merupakan hasil dari tampilan pada LCD. Jika LCD pada alat menampilkan hasil dari pemindaian, maka hasil tersebut akan dikatakan *State D* berhasil membaca sidik jari. Sedangkan, jika tidak terbaca, maka LCD tidak menampilkan apa – apa. Pada pengujian nomor 6, hasil dari pengujian tidak terbaca, dikarenakan sidik jari pengguna basah akibat keringat sehingga tidak dapat dibaca oleh sensor. Dari hasil pengujian ini didapatkan nilai kesuksesan sistem sebesar 90%.

### 4. Pengujian aksi *command* untuk menyalakan motor

Seperti yang dapat dilihat pada Tabel 6.5, bahwa pengujian pada aksi *command* untuk menjalankan *State E* berhasil. Dalam 10 kali pengujian dengan menggunakan *command* untuk menyalakan motor didapatkan tingkat kesuksesan sistem sebesar 100%.

## 6.1.3 Pengujian *State B1* : Cek *GPS*

Pada pengujian *State B1* : Cek *GPS* akan dilakukan satu kali pengujian berdasarkan satu aksi yang dilakukan oleh *State B1*, yaitu *time out* selama 1s atau *delay* 1s. Hasil dari pengujian ini akan ditampilkan dalam bentuk tabel dan pengujian ini diuji sebanyak 10 kali.

### 6.1.3.1 Tujuan

Tujuan dari pengujian *state B1* ini untuk mendapatkan hasil kinerja dari modul *GPS* yang digunakan serta modul SIM900A sebagai media komunikasi pengiriman data.

### 6.1.3.2 Prosedur

Kondisi motor dalam keadaan mati dan berada pada posisi yang sama pada satu tempat, yaitu garasi. Pengujian dimulai dari jam 02.51 pagi.

### 6.1.3.3 Hasil

**Table 6.6 Hasil pengujian dengan time out 1s**

Pengujian	Selisih Waktu(s)	Waktu pengguna meminta data(jam)	Waktu data diterima(jam)	Data lokasi yang diterima
1	7.16	02.51	02.51	-7.934756, 112.636960
2	8.78	02.51	02.52	-7.934756, 112.636960
3	6.72	02.52	02.52	-7.934756, 112.636960
4	11.70	02.52	02.53	-7.934733, 112.637004*

Pengujian	Selisih Waktu(s)	Waktu pengguna meminta data(jam)	Waktu data diterima(jam)	Data lokasi yang diterima
5	7.29	02.53	02.53	-7.934756, 112.636960
6	8.91	02.53	02.53	-7.934756, 112.636960
7	7.27	02.53	02.54	-7.934756, 112.637005*
8	7.92	02.54	02.54	-7.934756, 112.636960
9	6.06	02.54	02.54	-7.934756, 112.636960
10	6.48	02.55	02.55	-7.934756, 112.636960

#### 6.1.3.4 Analisis

Pada pengujian *State B1* yaitu cek *GPS* dengan aksi time out 1 detik ini dilakukan bersamaan pada pengujian *State A* dengan aksi memanggil *command* cek *GPS*, sehingga data pada lokasi yang didapatkan sama. Hasil dari pengujian ini ditunjukkan pada tabel di atas. Rata – rata selisih waktu dari data dikirim hingga diterima adalah 7.829 detik. Terdapat faktor dari luar yang menyebabkan data yang dikirim hingga data diterima memerlukan waktu melebihi time out 1 detik yaitu sinyal dari pengiriman SMS. Berdasarkan hasil yang telah diuji didapatkan kesuksesan sistem dalam mengirimkan data lokasi sebesar 100% dikarenakan ketika pengguna meminta data lokasi, sistem mengirimkan data lokasi.

#### 6.1.4 Pengujian *State B2* : Cek *GPS*

Pada pengujian *State B2* : Cek *GPS* akan dilakukan dua pengujian berdasarkan dua aksi yang dilakukan oleh *State B2*, yaitu aksi pada saat posisi berubah dan posisi tak berubah. Hasil dari pengujian ini akan ditampilkan dalam bentuk tabel dan pengujian ini diuji sebanyak 10 kali setiap aksi.

##### 6.1.4.1 Tujuan

Tujuan dari pengujian *state B2* ini untuk mendapatkan hasil kinerja dari modul *GPS* yang digunakan serta modul SIM900A sebagai media komunikasi pengiriman data.

##### 6.1.4.2 Prosedur

1. Pengujian aksi ketika posisi berubah

Kondisi awal motor mati dan berpindah lokasi dengan waktu time out selama 30 detik dan setiap 30 detik yang telah dihitung pada *stopwatch* motor akan berhenti sampai data lokasi diterima, kemudian ketika data telah diterima akan dilakukan pengujian kembali.

## 2. Pengujian aksi ketika posisi tidak berubah

Kondisi awal motor mati dan tidak berpindah tempat. Dan dilakukan pengujian dengan waktu time out selama 30s. Dimulai pada jam 3.55 pagi hingga 5 menit berikutnya.

### 6.1.4.3 Hasil

## 1. Pengujian aksi ketika posisi berubah

**Table 6.7 Hasil Pengujian Ketika Posisi Motor Berubah**

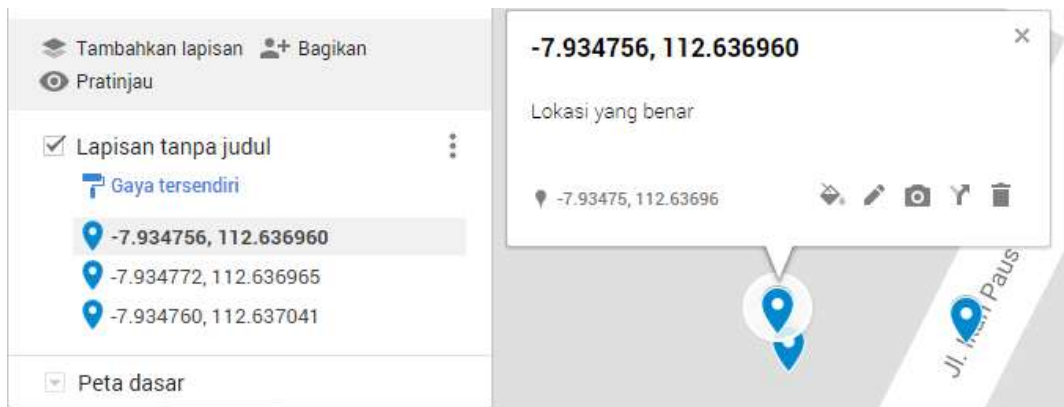
Pengujian	Data lokasi yang diterima
1	Mengirim ke pengguna data lokasi -7.934872, 112.636978
2	Mengirim ke pengguna data lokasi -7.934914, 112.636943
3	Mengirim ke pengguna data lokasi -7.934996, 112.636900
4	Mengirim ke pengguna data lokasi -7.935073, 112.636857
5	Mengirim ke pengguna data lokasi -7.935158, 112.636814
6	Mengirim ke pengguna data lokasi -7.935256, 112.636768
7	Mengirim ke pengguna data lokasi -7.935304, 112.636741
8	Mengirim ke pengguna data lokasi -7.935389, 112.636701
9	Mengirim ke pengguna data lokasi -7.935455, 112.636658
10	Mengirim ke pengguna data lokasi -7.935551, 112.636626

## 2. Pengujian aksi ketika posisi tidak berubah

**Table 6.8 Hasil Pengujian Ketika Posisi Motor Tidak Berubah**

Pengujian	Waktu Time out	Waktu data diterima(jam)	Data lokasi yang diterima
1	00:30.42	-	Tidak ada data yang diterima
2	01:00.80	-	Tidak ada data yang diterima
3	01:32.85	03.57	-7.934772, 112.636965
4	02:04.45	-	Tidak ada data yang diterima
5	02:35.29	-	Tidak ada data yang diterima
6	03:06.55	-	Tidak ada data yang diterima
7	03:37.37	-	Tidak ada data yang diterima
8	04:07.41	-	Tidak ada data yang diterima
9	04:41.64	04.00	-7.934760, 112.637041

10	05:11.86	-	Tidak ada data yang diterima
----	----------	---	------------------------------



**Gambar 6.5 Lokasi Hasil Pengujian State B2 Aksi dua**

#### **6.1.4.4 Analisis**

##### **1. Pengujian aksi ketika posisi berubah**

Pada pengujian *State B2* yaitu cek *GPS* dengan aksi ketika motor mati dan berpindah lokasi atau posisi ini dilakukan bersamaan pada pengujian *State A* dengan aksi time out 30s, sehingga data pada lokasi yang didapatkan sama. Pada pengujian ini didapatkan hasil kesuksesan sistem sebesar 100% dikarenakan 10 kali pengujian, semua data lokasi diterima oleh pengguna.

##### **2. Pengujian aksi ketika posisi tidak berubah**

Seperti pengujian sebelumnya pada pengujian *State a* dengan aksi *command* memanggil lokasi kondisi data *GPS* yang diterima ini terkadang mengalami perubahan dikarenakannya faktor dari luar, seperti contohnya adalah kondisi cuaca dan tempat pembacaan *GPS* yang berada dalam rumah. Namun, dari data yang diuji lokasi yang berubah tidak jauh dari lokasi sebenarnya. Pada pengujian ini tingkat kesuksesan sistem yang didapat sebesar 80%. Gambar lokasi dapat dilihat pada Gambar 6.5.

#### **6.1.5 Pengujian State C2 : Kirim Notif Lokasi**

Pada pengujian *State C2* : Kirim Notif Lokasi akan dilakukan satu kali pengujian berdasarkan satu aksi yang dilakukan oleh *State C2*, yaitu aksi pada saat time out 30s. Hasil dari pengujian ini akan ditampilkan dalam bentuk tabel dan pengujian ini diuji sebanyak 10 kali setiap aksi.

##### **6.1.5.1 Tujuan**

Tujuan dari pengujian *state C2* ini untuk mendapatkan hasil kinerja dari modul SIM900A sebagai media komunikasi pengiriman data.

#### 6.1.5.2 Prosedur

Kondisi motor dalam keadaan mati dan berpindah tempat, waktu pengujian diambil pada jam 03.24 pagi dan setiap 30 detik yang telah dihitung pada *stopwatch* motor akan berhenti sampai data lokasi diterima, kemudian ketika data telah diterima akan dilakukan pengujian kembali.

#### 6.1.5.3 Hasil

**Table 6.9 Hasil Pengujian aksi time out 30 s**

Pengujian	Selisih waktu(s)	Waktu data diterima(jam)	Data lokasi yang diterima
1	38.75	03.24	-7.934872, 112.636978
2	36.36	03.25	-7.934914, 112.636943
3	35.70	03.25	-7.934996, 112.636900
4	39.05	03.26	-7.935073, 112.636857
5	36.13	03.27	-7.935158, 112.636814
6	36.73	03.27	-7.935256, 112.636768
7	36.67	03.28	-7.935304, 112.636741
8	36.23	03.28	-7.935389, 112.636701
9	35.90	03.29	-7.935455, 112.636658
10	34.88	03.30	-7.935551, 112.636626

#### 6.1.5.4 Analisis

Pada pengujian C2 ini dilakukan bersamaan dengan pengujian pada *State A* dengan aksi time out 30s. Sehingga data yang dihasilkan sama. Pada pengujian ini didapatkan tingkat kesuksesan sistem sebesar 100% dikarenakan pengujian 10 kali, semua data lokasi dapat diterima oleh pengguna.

#### 6.1.6 Pengujian *State D* : Cek Fingerprint

Pada pengujian *State D* : Cek Fingerprint akan dilakukan dua kali pengujian berdasarkan dua aksi yang dilakukan oleh *State D*, yaitu aksi ketika sidik jari pengguna benar dan ketika sidik jari pengguna salah. Hasil dari pengujian ini akan ditampilkan dalam bentuk tabel dan pengujian ini diuji sebanyak 10 kali setiap aksi.

##### 6.1.6.1 Tujuan

Tujuan dari pengujian *state D* ini untuk mendapatkan hasil kinerja dari sensor sidik jari sebagai media pemindaian sidik jari pengguna.

#### 6.1.6.2 Prosedur

1. Pengujian aksi ketika sidik jari pengguna benar

Pada pengujian ini digunakan sidik jari penulis yang sebelumnya telah tersimpan.

2. Pengujian aksi ketika sidik jari pengguna salah

Pada pengujian ini digunakan sidik jari dari 3 orang berbeda. Untuk 2 orang melakukan 3 kali pengujian dan 1 orang melakukan 4 pengujian.

#### 6.1.6.3 Hasil

1. Pengujian aksi ketika sidik jari pengguna benar

**Table 6.10 Hasil Pengujian Sidik Jari Pengguna Benar**

Pengujian	Hasil
1	Mesin Nyala
2	Mesin Nyala
3	Mesin Nyala
4	Mesin Nyala
5	Mesin Nyala
6	Mesin Nyala
7	Mesin Nyala
8	Mesin Nyala
9	Mesin Nyala
10	Mesin Nyala

2. Pengujian aksi ketika sidik jari pengguna salah

**Table 6.11 Hasil Pengujian Sidik Jari Pengguna Salah**

Pengujian	Hasil
1	Mesin Tetap Mati
2	Mesin Tetap Mati
3	Mesin Tetap Mati
4	Mesin Tetap Mati
5	Mesin Tetap Mati
6	Mesin Tetap Mati
7	Mesin Tetap Mati

8	Mesin Tetap Mati
9	Mesin Tetap Mati
10	Mesin Tetap Mati

#### 6.1.6.4 Analisis

##### 1. Pengujian aksi ketika sidik jari pengguna benar

Seperti yang dapat dilihat pada tabel di atas, bahwa pengujian pada aksi ketika sidik jari pengguna benar berhasil. Jadi ketika pengguna yang memiliki sidik jari yang benar dan melakukan pemindaian didapatkan motor menyala. Dari 10 kali pengujian dengan menggunakan sidik jari untuk menyalakan motor dengan sidik jari yang benar, semuanya berhasil. Sehingga tingkat kesuksesan sistem pada pengujian ini sebesar 100%.

##### 2. Pengujian aksi ketika sidik jari pengguna salah

Pada tabel di atas menunjukkan bahwa sensor *fingerprint* mampu membaca sidik jari dari 3 orang dan dapat membandingkannya dengan sidik jari yang sebelumnya telah tersimpan. Jadi ketika pengguna yang memiliki sidik jari yang salah dan melakukan pemindaian didapatkan motor dalam kondisi tetap mati. Dari 10 kali pengujian dengan menggunakan sidik jari untuk menyalakan motor dengan sidik jari yang salah, semuanya berhasil. Sehingga tingkat kesuksesan sistem pada pengujian ini sebesar 100%.

#### 6.1.7 Pengujian State E : Mesin Nyala

Pada pengujian *State E* : Mesin nyala akan dilakukan dua kali pengujian berdasarkan dua aksi yang dilakukan oleh *State E*, yaitu aksi ketika ada *command* cek *GPS* dan ada *command* untuk mematikan mesin. Hasil dari pengujian ini akan ditampilkan dalam bentuk tabel dan pengujian ini diuji sebanyak 10 kali setiap aksi.

##### 6.1.7.1 Tujuan

Tujuan dari pengujian *state E* ini untuk mendapatkan hasil kinerja dari modul *GPS* dan kinerja dari modul *SIM900A* sebagai komunikasi pada sistem, serta kinerja pada relay.

##### 6.1.7.2 Prosedur

##### 1. Pengujian aksi *command* cek *GPS*

Kondisi motor dalam keadaan menyala dan motor berpindah - pindah lokasi. Kemudian dilakukan pengujian dengan memanggil *command* cek *GPS*.

##### 2. Pengujian aksi *command* mematikan mesin

Kondisi mesin dalam keadaan menyala dan motor berada pada satu lokasi yang tetap. Kemudian dilakukan pengujian dengan memanggil *command*



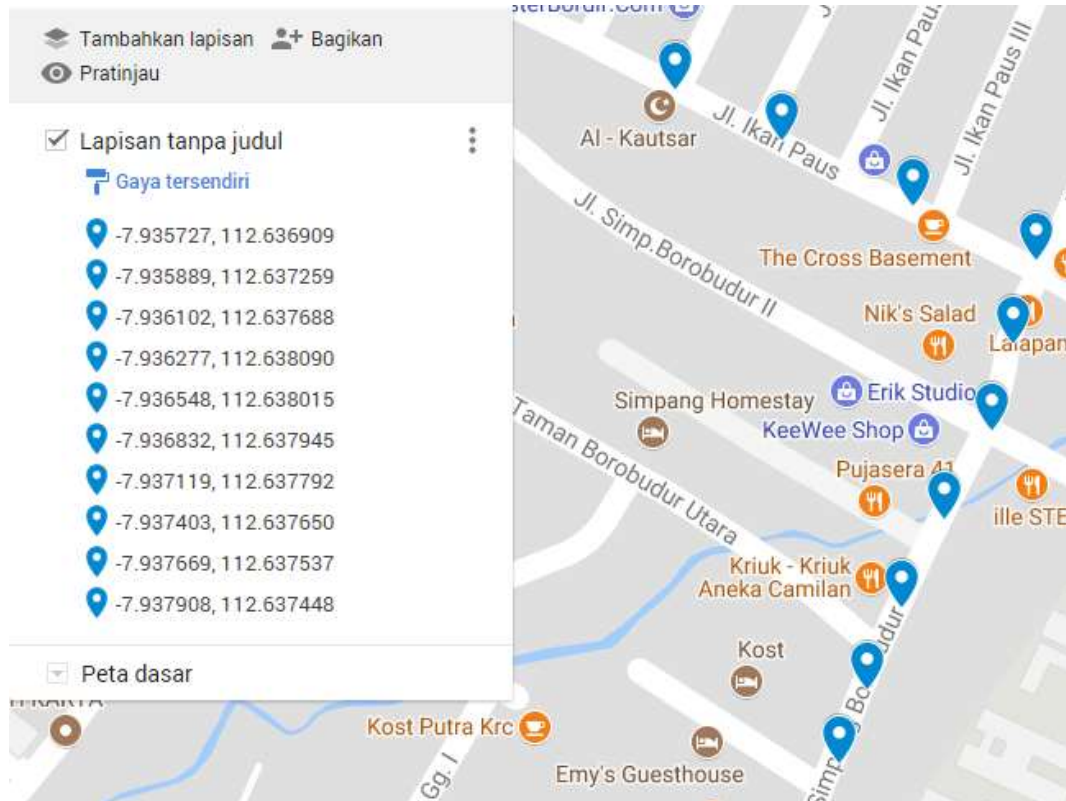
mematikan mesin. Setiap setelah mematikan mesin, mesin akan dinyalakan kembali.

### 6.1.7.3 Hasil

1. Pengujian aksi *command* cek *GPS*

**Table 6.12 Hasil Pengujian *Command* Cek *GPS***

Pengujian	Data lokasi yang diterima
1	Mengirim ke pengguna data lokasi -7.935727, 112.636909
2	Mengirim ke pengguna data lokasi -7.935889, 112.637259
3	Mengirim ke pengguna data lokasi -7.936102, 112.637688
4	Mengirim ke pengguna data lokasi -7.936277, 112.638090
5	Mengirim ke pengguna data lokasi -7.936548, 112.638015
6	Mengirim ke pengguna data lokasi -7.936832, 112.637945
7	Mengirim ke pengguna data lokasi -7.937119, 112.637792
8	Mengirim ke pengguna data lokasi -7.937403, 112.637650
9	Mengirim ke pengguna data lokasi -7.937669, 112.637537
10	Mengirim ke pengguna data lokasi -7.937908, 112.637448



**Gambar 6.6 Lokasi Berdasarkan Data Dari Tabel 6.11**

Lokasi pada google maps ditandai dengan tanda berwarna biru pada Gambar 6.11

## 2. Pengujian aksi *command* mematikan mesin

**Table 6.13 Hasil Pengujian *Command* Mematikan Mesin**

Pengujian	Hasil
1	Mesin Mati
2	Mesin Mati
3	Mesin Mati
4	Mesin Mati
5	Mesin Mati
6	Mesin Mati
7	Mesin Mati
8	Mesin Mati
9	Mesin Mati
10	Mesin Mati

#### 6.1.7.4 Analisis

##### 1. Pengujian aksi *command* cek *GPS*

Pada tabel di atas pengujian pertama merupakan lokasi awal dari motor yang kemudian jalan dan berpindah lokasi. Dari pengujian ini didapatkan hasil kesuksesan sistem sebesar 100% dikarenakan semua data lokasi dari 10 kali pengujian diterima oleh pengguna. Data lokasi dapat dilihat pada Gambar 6.11 untuk lebih jelasnya.

##### 2. Pengujian aksi *command* mematikan mesin

Pengujian dengan menggunakan *command* untuk mematikan mesin berhasil dengan hasil pengujian mesin pada motor, mati. Dalam 10 kali pengujian dengan menggunakan *command* untuk mematikan motor didapatkan tingkat kesuksesan sistem sebesar 100%.

#### 6.1.8 Pengujian *State B3* : Cek *GPS*

Pada pengujian *State B3* : Cek *GPS* akan dilakukan satu kali pengujian berdasarkan satu aksi yang dilakukan oleh *State B1*, yaitu time out selama 1s atau delay 1s. Hasil dari pengujian ini akan ditampilkan dalam bentuk tabel dan pengujian ini diuji sebanyak 10 kali.

##### 6.1.8.1 Tujuan

Tujuan dari pengujian *state B3* ini untuk mendapatkan hasil kinerja dari modul *GPS* yang digunakan serta modul SIM900A sebagai media komunikasi pengiriman data.

##### 6.1.8.2 Prosedur

Kondisi motor dalam keadaan nyala dan motor berpindah-pindah lokasi. Kemudian dilakukan *command* meminta lokasi. Pengujian dimulai dari jam 4.12 pagi.

##### 6.1.8.3 Hasil

Table 6.14 Hasil Pengujian Cek *GPS* Dengan Time Out 1s

Pengujian	Selisih waktu(s)	Waktu pengguna meminta data(jam)	Waktu data diterima(jam)	Data lokasi yang diterima
1	07.27	04:13	04:13	-7.935727, 112.636909
2	08.27	04:14	04:14	-7.935889, 112.637259

3	06.72	04:14	04:15	-7.936102, 112.637688
4	07.47	04:15	04:15	-7.936277, 112.638090
5	08.30	04:16	04:16	-7.936548, 112.638015
6	07.20	04:16	04:17	-7.936832, 112.637945
7	06.39	04:17	04:17	-7.937119, 112.637792
8	06.83	04:17	04:18	-7.937403, 112.637650
9	06.90	04:18	04:18	-7.937669, 112.637537
10	07.38	04:18	04:19	-7.937908, 112.637448

#### 6.1.8.4 Analisis

Pada pengujian *State* B3 yaitu cek *GPS* dengan aksi time out 1 detik ini dilakukan bersamaan pada pengujian *State* E dengan aksi memanggil *command* cek *GPS*, sehingga data pada lokasi yang didapatkan sama. Hasil dari pengujian ini yang ditunjukkan pada tabel di atas rata – rata selisih waktu dari waktu pengguna meminta data hingga data diterima adalah 7.273 detik. Pada pengujian ini didapatkan tingkat kesuksesan sistem sebesar 100% dikarenakan data lokasi yang diminta oleh pengguna dapat dikirim sistem kepada pengguna dengan berhasil

## 6.2 Analisis Pengujian

Setelah dilakukannya pengujian pada sistem, kemudian akan dilakukan analisis terhadap hasil yang telah diperoleh pada pengujian. Analisis pengujian akan dilakukan dengan dua cara, yaitu menganalisis setiap hasil berdasarkan pengujian pada *state* yang telah dilakukan sebelumnya pada setiap *state* dan analisis keberhasilan keseluruhan sistem. Pada analisis keberhasilan keseluruhan sistem akan dilakukan perhitungan yang berdasarkan tingkat kesuksesan sistem pada setiap *state* yang diuji sebanyak 13 jenis pengujian. Pada *state* A didapatkan 4 tingkat kesuksesan berdasarkan 4 jenis pengujian *state*, yaitu 80% pada pengujian aksi *command* memanggil lokasi, 100% pada pengujian aksi *time out* 30s, 90% pada pengujian pemindaian sidik jari dan 100% pada aksi *command* menyalakan motor. Pada pengujian *state* B1 didapatkan tingkat kesuksesan sebesar 100%. Pada *state* B2 didapatkan 2 tingkat kesuksesan berdasarkan 2 jenis pengujian *state*, yaitu 100% pada pengujian ketika motor berubah posisi dan 80% pada pengujian ketika motor tidak berubah posisi. Pada pengujian *state* C2 didapatkan tingkat kesuksesan sebesar 100%. Pada *state* D didapatkan 2 tingkat kesuksesan berdasarkan 2 jenis pengujian *state*, yaitu 100% pada pengujian ketika sidik jari benar dan 100% pada pengujian ketika sidik jari salah. Pada *state* E didapatkan 2 tingkat kesuksesan sistem berdasarkan 2 jenis pengujian *state*, yaitu 100% pada pengujian aksi *command* memanggil lokasi dan 100% pada pengujian aksi *command* mematikan motor. Serta, 100% tingkat kesuksesan pada pengujian *state* B3.

Dari semua tingkat kesuksesan yang didapat kemudian dijumlahkan yang menghasilkan nilai 1150%. Setelah semua tingkat kesuksesan sistem berdasarkan masing – masing pengujian dijumlahkan berikutnya dibagi dengan nilai 1300 yang merupakan nilai dari 13 pengujian dikalikan dengan 100%. Sehingga didapatkan hasil tingkat kesuksesan keseluruhan sistem sebesar 88,46%.

## BAB 7 PENUTUP

Bagian ini memuat kesimpulan dan saran terhadap penelitian. Kesimpulan dan saran disajikan secara terpisah, dengan penjelasan sebagai berikut:

### 7.1 Kesimpulan

Berdasarkan hasil perancangan sistem, implementasi sistem, pengujian, dan analisis sistem, maka penulis menyimpulkan:

1. Untuk merancang dan mengimplementasikan sebuah sistem embedded yang dapat mengamati dan memantau kendaraan motor dengan menggunakan *state machine* dibutuhkan perangkat keras seperti sensor sidik jari, modul SIM900A, modul *GPS* SKM53, modul relay 1 *channel*, *power supply* berupa aki dan LCD 16x4. Kebutuhan perangkat lunak yaitu *ApplInverter*.
2. Untuk pengiriman dan penerimaan data lokasi menggunakan komunikasi GSM dengan memanfaatkan *SMS-Gateway* menunjukkan kecepatan waktu pengiriman yang cukup dapat dikatakan singkat. Karena, rata – rata waktu yang dibutuhkan data untuk dikirim dan diterima relatif singkat. Berdasarkan hasil dari pengujian yang telah diuji untuk waktu pengiriman dan penerimaan data lokasi didapatkan 3 waktu rata – rata berdasarkan 3 *state* yang bekerja pada pengiriman dan penerimaan data lokasi, yaitu *state* B2 6.64 detik, *state* B1 7.829 detik dan *state* B3 7.273 detik.
3. Tingkat keberhasilan yang didapatkan oleh sistem dengan menerapkan *State Machine* dapat dikatakan berhasil dengan nilai tingkat keberhasilan keseluruhan sistem adalah 88,46%. Dikarenakan semua sensor dan modul yang digunakan pada sistem dapat berjalan dengan baik. Meskipun pada beberapa pengujian terdapat kesalahan sistem yang menyebabkan hasil tidak sesuai dengan yang diinginkan. Namun, jika dibandingkan dengan proses yang berhasil masih dapat dikatakan sistem berjalan dengan baik.

### 7.2 Saran

Berikut ini merupakan saran yang diberikan penulis dengan tujuan agar penelitian ini dapat lebih dikembangkan:

1. Untuk pengembangan lebih lanjut, dapat menggunakan metode pembacaan lokasi lainnya agar proses pembacaan lokasi menghasilkan data yang lebih akurat dan konstan ketika berada pada 1 tempat.
2. Pada pengembangan berikutnya, bisa menggunakan module *GPS* yang lebih mampu untuk membaca lokasi ketika berada di dalam ruangan.
3. Untuk penelitian selanjutnya dapat menerapkan metode selain *State Machine* pada sistem keamanan motor.

## DAFTAR PUSTAKA

- Ahmad, F., Santo, T., Mohammad, A., & Mirza, T. (2014). *Sistem Keamanan Motor Melalui Short Message Service Menggunakan AVR Mikrokontroler ATmega8*. STIMIK Raharja.
- Al-Fedaghi, S., & Alazmi, A. (2014). *A New Approach to Specification of The Behaviour of Embedded System*. Kuwait: Kuwait University.
- Arduino. (2017). *Arduino Uno Rev3*. Retrieved from Arduino: <https://store.arduino.cc/usa/arduino-uno-rev3>
- Baradja, M. A. (2016). *Rancang Bangun Sistem Keamanan Kendaraan Bermotor Berbasis GPS*. Universitas Airlangga.
- Beman, S., Steven, F., & Liawatimena, S. (2011). *Perancangan Sistem Keamanan Sepeda Motor Dengan Sistem Sidik Jari*. Universitas Binus.
- Binanto, I. (2015). *Analisa Metode Classic Life Cycle (Waterfall) Untuk Pengembangan Perangkat Lunak Multimedia*. Yogyakarta: Universitas Sanata Dharma.
- Brownlee, J. (2010). *Finite State Machine (FSM)*. Retrieved from AI-depot: <http://ai-depot.com/FiniteStateMachines/FSM.html>
- Kodavati, B., Raju, V., Rao, S. S., Prabu, A., Rao, T. A., & Narayana, D. Y. (2012). GSM AND GPS BASE VEHICLE LOCATION AND TRACKING SYSTEM. *International Journal of Engineering Research and Applications*, 616-625.
- Madhu, R. (2016). GSM/GPS based Device Switching with Fingerprint Module Integration using Arduino. *International Journal of Computer Applications*, 28-32.
- Salim, P., & salim, Y. (2002). *Kamus Bahasa Indonesia Kontemporer*. Jakarta: Modern English Press.
- Setiawan, I. (2006). *Perancangan Software Embedded System Berbasis FSM*. Diponegoro: Universitas Diponegoro.
- Singh, P., Sethi, T., Biswal, B., & Pattanayak, S. (2015). *A Smart Anti-Theft System for Vehicle Security*. National Institute of Technology Rourkela.
- Sommerville, I. (2011). *Software Engineering*. Scotland: Pearson.
- Statistik, B. P. (2014). *Statistik Kriminal 2014*. Retrieved September 11, 2017, from [https://www.bappenas.go.id/files/data/Politik\\_Hukum\\_Pertahanan\\_dan\\_Keamamanan/Statistik%20Kriminal%202014.pdf](https://www.bappenas.go.id/files/data/Politik_Hukum_Pertahanan_dan_Keamamanan/Statistik%20Kriminal%202014.pdf)
- Vigneshwaran, K., Sumithra, S., & Janani, R. (2015). An Intelligent Tracking System Based on GSM and GPS Using Smartphones. *International Journal of Engineering Research and Applications*, 3897-3903.

## LAMPIRAN A *Source Code* Program

### A.1 Program Utama

	Main Program
1	#include <Adafruit_Fingerprint.h>
2	#include <LiquidCrystal.h>
3	#include <SoftwareSerial.h>
4	#define DEBUG true
5	#include <TinyGPS.h>
6	#include <SoftwareSerial.h>
7	#include "Fsm.h"
8	#define CONDITION 0
9	SoftwareSerial Finger(2, 3);
10	LiquidCrystal lcd(12, 11, A3, A2, A1, A0);
11	SoftwareSerial GPS(4,5);
12	SoftwareSerial SIM900A(9,10);
13	TinyGPS GPS;
14	long lat, lon;
15	float LAT, LON;
16	
17	String strTemp,strTemp1,Lat_sebelumnya,Lon_sebelumnya;
18	char buf[16],buff[16];
19	int notconnect=1;
20	int latitude,longtitude,limit;
21	char a;
22	int kondisi=0,id=80;
23	Adafruit_Fingerprint                      finger                      =
24	Adafruit_Fingerprint(&Finger);
25	////////////////////////////////////
26	/
27	void getGPS();
28	bool feedGPS();
29	void skm53();
30	void SendMessage();
31	void message_GPS();
32	void RecieveMessage();
33	void delete_mesage();
34	uint8_t getFingerprintID();
35	int getFingerprintIDez();
36	void StateMachine();
37	void Motor_ON();
38	void Motor_OFF();
39	////////////////////////Inisialisasi                      State
40	Machine////////////////////////////////
41	State State_A(&Motor_OFF, NULL, NULL);
42	State State_B1(&skm53, NULL, NULL);
43	State State_C1(&SendMessage, NULL, NULL);
44	State State_B2(&skm53, NULL, NULL);
45	State State_C2(&SendMessage, NULL, NULL);



```

46 State State_D(&getFingerprintIDez, NULL, NULL);
47 State State_E(&Motor_ON, NULL, NULL);
48 State State_B3(&skm53, NULL, NULL);
49 State State_C3(&SendMessage, NULL, NULL);
50
51 Fsm fsm_A1(&State_A);
52 Fsm fsm_A2(&State_A);
53 Fsm fsm_A3(&State_A);
54 Fsm fsm_A4(&State_A);
55 Fsm fsm_B1(&State_B1);
56 Fsm fsm_B2(&State_B2);
57 Fsm fsm_C2(&State_C2);
58 Fsm fsm_D1(&State_D);
59 Fsm fsm_D2(&State_D);
60 Fsm fsm_E1(&State_E);
61 Fsm fsm_E2(&State_E);
62 Fsm fsm_B3(&State_B3);
63 void setup()
64 {
65     Serial.begin(9600);
66     lcd.begin(16,4);
67     pinMode(8,OUTPUT);
68     fsm_A1.add_transition(&State_A, &State_B1, NULL, NULL);
69     fsm_A2.add_timed_transition(&State_A, &State_B2, 30000,
70 NULL);
71     fsm_A3.add_transition(&State_A, &State_D, NULL, NULL);
72     fsm_A4.add_transition(&State_A, &State_E, NULL, NULL);
73     fsm_B1.add_timed_transition(&State_B1, &State_C1, 1000,
74 NULL);
75     fsm_B2.add_timed_transition(&State_B2, &State_C2, 1000,
76 NULL);
77     fsm_C2.add_timed_transition(&State_C2, &State_B2, 30000,
78 NULL);
79     fsm_D1.add_transition(&State_D, &State_E, NULL, NULL);
80     fsm_D2.add_transition(&State_D, &State_A, NULL, NULL);
81     fsm_E1.add_timed_transition(&State_E, &State_B3, 1000,
82 NULL);
83     fsm_E2.add_timed_transition(&State_E, &State_A, 1000,
84 NULL);
85     fsm_B3.add_timed_transition(&State_B3, &State_C3, 1000,
86 NULL);
87     while (!Serial); // For Yun/Leo/Micro/Zero/...
88     delay(100);
89     Serial.println("\n\nAdafruit finger detect test");
90     // set the data rate for the sensor serial port
91     finger.begin(57600);
92
93     if (finger.verifyPassword()) {
94         Serial.println("Found fingerprint sensor!");
95     } else {
96         Serial.println("Did not find fingerprint sensor :(");

```

97	while (1) { delay(1); }
98	}
99	SIM900A.begin(19200);
100	SIM900A.println("AT+CMGF=1");
101	delay(1000);
102	SIM900A.println("AT+CNMI=2,2,0,0,0"); // AT Command to
103	receive a live SMS
104	delay(1000);
105	pinMode(8,OUTPUT);
106	if (notconnect==1) {
107	GPS.begin(9600);
108	skm53();
109	}
110	finger.getTemplateCount();
111	Serial.print("Sensor contains ");
112	Serial.print(finger.templateCount); Serial.println("
113	templates");
114	Serial.println("Waiting for valid finger...");
115	}
116	
117	void loop() // run over and over again
118	{
119	StateMachine();
120	delay(50); //don't need to run this at full
121	speed.
122	}

## A.2 Program Aktuator

	Program Aktuator
1	void Motor_ON(){
2	digitalWrite(8,HIGH);
3	}
4	void Motor_OFF(){
5	digitalWrite(8,LOW);
6	}

## A.3 Program GPS

	Program GPS
1	void getGPS(){
2	bool newdata = false;
3	unsigned long start = millis();
4	// Every 1 seconds we print an update
5	while (millis() - start < 1000)
6	{
7	if (feedGPS()){
8	newdata = true;
9	}

```

10     }
11     if (newdata)
12     {
13         GPSdump(GPS);
14     }
15 }
16
17 bool feedGPS(){
18     while (GPS.available())
19     {
20         if (GPS.encode(GPS.read()))
21             return true;
22     }
23     return 0;
24 }
25
26 void GPSdump(TinyGPS &GPS)
27 {
28     //byte month, day, hour, minute, second, hundredths;
29     GPS.get_position(&lat, &lon);
30     LAT = lat;
31     LON = lon;
32     {
33         feedGPS(); // If we don't feed the GPS during this
34         long routine, we may drop characters and get checksum
35         errors
36     }
37 }
38 void skm53(){
39     GPS.listen();
40     long lat, lon;
41     unsigned long fix_age, time, date, speed, course;
42     unsigned long chars;
43     unsigned short sentences, failed_checksum;
44
45     // retrieves +/- lat/long in 100000ths of a degree
46     GPS.get_position(&lat, &lon, &fix_age);
47
48     getGPS();
49     strTemp = dtostrf(LAT/1000000, 4, 4, buf);
50     strTemp1 = dtostrf(LON/1000000, 4, 4, buff);
51     latitude=atoi(buf);
52     longitude=atoi(buff);
53     lcd.setCursor(0, 0);
54     lcd.print("Location");
55     //Serial.print("Location (in degrees, works with Google
56     Maps): ");
57     lcd.setCursor(0, 1);
58     lcd.print(strTemp);
59     lcd.print(",");
60     lcd.print(strTemp1);

```

61	//Serial.print(strTemp);
62	//Serial.print(", ");
63	//Serial.println(strTemp1);
	}

#### A.4 Program *State Machine*

	<b>Program <i>State Machine</i></b>
1	void StateMachine() {
2	getFingerprintIDez();
3	RecieveMessage();
4	skm53();
5	fsm_A2.run_machine();
6	
7	if(strTemp!=Lat_sebelumnya  strTemp1!=Lon_sebelumnya){fsm_
8	B2.run_machine();fsm_C2.run_machine();}
9	if(finger.fingerID==id){kondisi=1;}
10	
11	if(SIM900A.available()=='A'){fsm_A1.run_machine();fsm_B1.r
12	un_machine();}
13	
14	if(SIM900A.available()=='B'){fsm_A4.run_machine();kondisi=
15	1;}
16	if(kondisi==0){fsm_D2.run_machine();}
17	while(kondisi==1){
18	fsm_A3.run_machine();
19	fsm_D1.run_machine();
20	RecieveMessage();
21	if(SIM900A.available()=='A'){
22	fsm_E1.run_machine();
23	fsm_B3.run_machine();
24	}
25	if(SIM900A.available()=='C'){
26	fsm_E2.run_machine();
27	kondisi=0;
28	return;
29	}
30	}
31	
32	Lat_sebelumnya=strTemp;
33	Lon_sebelumnya=strTemp1;
34	
35	}

#### A.5 Program Sim900A

	<b>Program SIM900A</b>
1	void SendMessage()
2	{

3	lcd.clear();
4	delay(250);
5	lcd.setCursor(0, 0);
6	lcd.print ("Mengirim SMS");
7	SIM900A.println("AT+CMGF=1"); //Sets the GSM Module
8	in Text Mode
9	delay(1000); // Delay of 1000 milli seconds or 1 second
10	//Serial.println ("Set SMS Number");
11	SIM900A.println("AT+CMGS=\"+6281333021364\\\"\\r\"); //
12	Replace with your mobile number
13	delay(1000);
14	
15	}
16	void message_GPS(){
17	//Serial.println ("Set SMS Content");
18	SIM900A.println("Lokasi Motor");// The SMS text you want
19	to send
20	SIM900A.println(strTemp);
21	SIM900A.print(",");// The SMS text you want to send
22	SIM900A.println(strTemp1);
23	
24	delay(100);
25	lcd.setCursor(0, 1);
26	lcd.println ("Finish");
27	SIM900A.println((char)26);// ASCII code of CTRL+Z
28	delay(1000);
29	//Serial.println (" ->SMS Selesai dikirim");
30	}
31	void RecieveMessage()
32	{
33	//Serial.println ("SIM900A Membaca SMS");
34	delay (1000);
35	SIM900A.println("AT+CNMI=2,2,0,0,0"); // AT Command to
36	receive a live SMS
37	delay(1000);
38	//Serial.write (" ->Unread SMS Selesai dibaca");
39	}
40	void delete_mesage(){
41	delay(1000);
42	//Did not help
43	SIM900A.print ("AT+CMGDA=");
44	SIM900A.println("DEL ALL");
45	delay(500);
46	}

## A.6 Program Fingerprint

	Main Program
1	uint8_t getFingerprintID() {
2	uint8_t p = finger.getImage();

```

3      switch (p) {
4          case FINGERPRINT_OK:
5              Serial.println("Image taken");
6              break;
7          case FINGERPRINT_NOFINGER:
8              Serial.println("No finger detected");
9              return p;
10         case FINGERPRINT_PACKETRECEIVEERR:
11             Serial.println("Communication error");
12             return p;
13         case FINGERPRINT_IMAGEFAIL:
14             Serial.println("Imaging error");
15             return p;
16         default:
17             Serial.println("Unknown error");
18             return p;
19     }
20
21     // OK success!
22
23     p = finger.image2Tz();
24     switch (p) {
25         case FINGERPRINT_OK:
26             Serial.println("Image converted");
27             break;
28         case FINGERPRINT_IMAGEMESS:
29             Serial.println("Image too messy");
30             return p;
31         case FINGERPRINT_PACKETRECEIVEERR:
32             Serial.println("Communication error");
33             return p;
34         case FINGERPRINT_FEATUREFAIL:
35             Serial.println("Could not find fingerprint
36 features");
37             return p;
38         case FINGERPRINT_INVALIDIMAGE:
39             Serial.println("Could not find fingerprint
40 features");
41             return p;
42         default:
43             Serial.println("Unknown error");
44             return p;
45     }
46
47     // OK converted!
48     p = finger.fingerFastSearch();
49     if (p == FINGERPRINT_OK) {
50         Serial.println("Found a print match!");
51     } else if (p == FINGERPRINT_PACKETRECEIVEERR) {
52         Serial.println("Communication error");
53         return p;

```

```

54     } else if (p == FINGERPRINT_NOTFOUND) {
55         Serial.println("Did not find a match");
56         return p;
57     } else {
58         Serial.println("Unknown error");
59         return p;
60     }
61
62     // found a match!
63     Serial.print("Found          ID          #");
64     Serial.print(finger.fingerID);
65     Serial.print("          with          confidence          of          ");
66     Serial.println(finger.confidence);
67
68     return finger.fingerID;
69 }
70
71 // returns -1 if failed, otherwise returns ID #
72 int getFingerprintIDez() {
73     uint8_t p = finger.getImage();
74     if (p != FINGERPRINT_OK) return -1;
75
76     p = finger.image2Tz();
77     if (p != FINGERPRINT_OK) return -1;
78
79     p = finger.fingerFastSearch();
80     if (p != FINGERPRINT_OK) return -1;
81
82     // found a match!
83     Serial.print("Found          ID          #");
84     Serial.print(finger.fingerID);
85     Serial.print("          with          confidence          of          ");
86     Serial.println(finger.confidence);
87     return finger.fingerID;
88 }

```